

MOBILE ROBOT INITIALIZATION IN A BUILDING BASED ON A BUILDING INFORMATION MODEL (BIM) OF THE BUILDING

CROSS-REFERENCE TO RELATED APPLICATION

- 5 **[0001]** This application claims the benefit of priority of Singapore Patent Application No. 10202250807D filed on 23 August 2022 and Singapore Patent Application No. 10202251597W filed on 2 November 2022, the contents of which being hereby incorporated by reference in their entirety for all purposes.

10

TECHNICAL FIELD

[0002] The present invention generally relates to mobile robot initialization in a building based on a building information model (BIM) of the building, and more particularly, relates to a method and a system for mobile robot initialization in a building based on a BIM of the building, and a mobile robot with automated mobile robot initialization functionality.

15

BACKGROUND

[0003] There are a wide range of practical applications in which mobile robot initialization in a building may be desired or required. For example, the construction industry plays a pivotal role in the economic growth of many countries. Construction works are inherently labour-intensive, inefficient and rely heavily on manual processes, which are prone to instrumental and human errors and fatigue. In recent years, there has been a rapid development of robotic solutions for construction automation that brings tremendous improvements in productivity, quality and other aspects of construction work. Various mobile robotic and mechatronic systems were developed to perform certain tasks in construction automation through navigating around the construction site. For example, a waste recycling robot was developed to inspect the entire construction environment and detect the nails and screw waste. The construction monitoring robots mounted with scanning sensors and cameras were developed to navigate in the working environment to automate the construction inspection either for in-progress monitoring or post-construction quality and defects check.

20 **[0004]** In the robot navigation stack, the mobile robot initialization process may be important. For example, as the mobile robot may be deployed at an unknown location in a working space, the mobile robot's initial position in the map must first be determined so that desired path(s) to target location(s) can be planned. In addition, if the mobile robot loses its

location while navigating, re-initialization is required to recover or rectify its position on the map to resume navigation. Any wrong initialization can cause failure in robot pose tracking and thus prevent the mobile robot from performing the planned tasks.

[0005] In indoor or GPS-challenged (e.g., GPS-denied) environments (e.g., construction environments) where spaces are concealed by building structures, acquisition of mobile robot location information is challenging. For most of the commercial products of indoor mobile robots, the robot initialization is performed manually by the robot operator. Such initialization is usually performed by expert users, such as robot engineers since it requires certain knowledge of the robot navigation system. To achieve fully automated robot navigation (e.g., in construction sites), an autonomous robot initialization system is needed to make the system easily operated by non-expert users from any position. Some automated indoor positioning systems were proposed to realize the localization of mobile devices based on various technologies, such as Wi-Fi, Bluetooth Low Energy (BLE), image processing and laser scan matching. Despite the extensive research in indoor positioning, these conventional methods or approaches are infrastructure-based or environment-dependent. For example, some conventional methods require the physical deployment of instruments for localization. Infrastructure-free methods usually rely on pre-scanning and mapping of each different environment. For example, for a building construction site, changes in scenes and structures for various construction blocks at different construction stages make it difficult and inefficient to apply such conventional methods. Moreover, the initialization performance is usually poor for highly repetitive environments. Therefore, mobile robot initialization in a building remains a challenge for various working/operating environments, such as in building construction sites where telecommunication infrastructure is not yet available, operating environments are highly repetitive and/or symmetric, and mobile robots are expected to work in changing environments, such as different construction blocks.

[0006] A need therefore exists to provide a method of mobile robot initialization in a building, and a system thereof, that seeks to overcome, or at least ameliorate, one or more deficiencies in conventional methods or approaches in mobile robot initialization in a building, and more particularly, to provide automated mobile robot initialization with enhanced or improved robustness or effectiveness, for example, capable of operating under a variety of working/operating environments, including challenging environments such as GPS-denied environments, highly repetitive and/or symmetric environments and/or changing working environments. It is against this background that the present invention has been developed.

SUMMARY

[0007] According to a first aspect of the present invention, there is provided a method of mobile robot initialization in a building based on a building information model (BIM) of the building, the method comprising:

obtaining, from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building;

detecting a plurality of building objects of the building from the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects;

matching the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and

estimating a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of detected building objects.

[0008] According to a second aspect of the present invention, there is provided a system for mobile robot initialization in a building based on a BIM of the building, the system comprising:

at least one memory; and

at least one processor communicatively coupled to the at least one memory and configured to:

obtain, from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building;

detect a plurality of building objects of the building from the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects;

match the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and

estimate a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of detected building objects.

[0009] According to a third aspect of the present invention, there is provided a computer program product, embodied in one or more non-transitory computer-readable storage mediums, comprising instructions executable by at least one processor to perform the method of mobile robot initialization in a building based on a BIM of the building according to the above-mentioned first aspect of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Embodiments of the present invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

FIG. 1 depicts a schematic flow diagram of a method of mobile robot initialization in a building based on a BIM of the building, according to various embodiments of the present invention;

FIG. 2 depicts a schematic block diagram of a system for mobile robot initialization in a building based on a BIM of the building, according to various embodiments of the present invention;

FIG. 3 depicts a schematic drawing of a mobile robot with automated mobile robot initialization capability or functionality, according to various embodiments of the present invention;

FIG. 4 depicts a schematic drawing showing an overview of a robot initialization system, according to various example embodiments of the present invention;

FIG. 5 depicts a sample BIM (digital representation) for one single floor in a residential building, according to various example embodiments of the present invention;

FIG. 6A depicts a 3D model of the example door element digitally represented in the BIM, according to various example embodiments of the present invention;

FIG. 6B, FIG. 6C and FIG. 6D show example geometric information, example location information and example semantic information of the example door element, respectively, according to various example embodiments of the present invention;

FIG. 7A depicts an example user interface for BIM information retrieval, according to various example embodiments of the present invention;

FIG. 7B depicts an example of the extended graphical programming interface in Dynamo, according to various example embodiments of the present invention;

FIG. 8 shows Table 1 including example BIM information of various building objects, according to various example embodiments of the present invention;

FIG. 9 shows example bounding box predictions with prior anchor boxes, according to various example embodiments of the present invention;

5 FIG. 10A shows example images collected from different sources for the image dataset for the object detector training, according to various example embodiments of the present invention;

FIG. 10B shows example building object detection results, according to various example embodiments of the present invention;

10 FIG. 11 depicts a schematic flow diagram of a method of mobile robot initialization in a building based on a BIM of the building, according to various example embodiments of the present invention;

FIG. 12 shows the relationships or transformation between different frames, including the BIM frame, the odometry frame, the robot frame, the Lidar frame, the camera frame and the image frame, according to various example embodiments of the present invention;

FIG. 13 depicts a schematic flow diagram of a method of filtering detected building objects according to various example embodiments of the present invention;

FIG. 14 shows the relationships between the locations of objects in the odometry frame and BIM frame, according to various example embodiments of the present invention;

20 FIG. 15 illustrates matching of three building objects, according to various example embodiments of the present invention;

FIG. 16 shows an object matching algorithm (Algorithm 1), according to various example embodiments of the present invention;

FIG. 17 shows example costs of matching detected object groups to corresponding BIM objects, according to various example embodiments of the present invention;

FIG. 18A illustrates robot localization using trilateration, according to various example embodiments of the present invention;

FIG. 18B illustrates the estimation of robot heading direction (or orientation), according to various example embodiments of the present invention;

30 FIG. 19 shows a robot local navigation algorithm (Algorithm 2), according to various example embodiments of the present invention;

FIG. 20 shows an example scenario that illustrates the robot local navigation, according to various example embodiments of the present invention;

FIG. 21 shows an overall online decision-making algorithm (Algorithm 3) for robot initialization (e.g., coarse initialization), according to various example embodiments of the present invention;

5 FIG. 22 depicts a schematic block diagram showing the multi-level positioning rectification, according to various example embodiments of the present invention;

FIGs. 23A and 23B depict schematic drawings illustrating example multi-level positioning rectifications, according to various example embodiments of the present invention;

FIG. 24 depicts an example mobile robot, according to various example embodiments of the present invention;

10 FIGs. 25A, 25B and 25C show example scenes of various testing environments, namely, a university building, residential block A and residential block B, according to various example embodiments of the present invention;

FIG. 26 shows a 3D view of the BIM model for the hallway of the university building, according to various example embodiments of the present invention;

15 FIG. 27 shows a 2D floor plan for the hallway of the university building, along with locations of various building objects thereat, according to various example embodiments of the present invention;

FIG. 28 shows Table 2 presenting BIM information of various building objects of the university building, according to various example embodiments of the present invention;

20 FIG. 29 shows various robot starting positions and captured scenes at the hallway of the university building, according to various example embodiments of the present invention;

FIG. 30A shows the relationships between BIM building object distances and the measured distance errors in experiments performed, according to various example embodiments of the present invention;

25 FIG. 30A shows the relationships between BIM building object distances and the measured distance errors in experiments performed, according to various example embodiments of the present invention;

30 FIG. 31 shows Table 3 presenting building object matching results for different robot starting positions at the hallway of the university building, according to various example embodiments of the present invention;

FIG. 32 depicts a graph showing variations in the number of feasible matches with the number of building objects detected, according to various example embodiments of the present invention;

FIG. 33 illustrates the process of filtering out wrong matching candidates for feasible matches for 3, 4, 5, 6 detected building objects, respectively, according to various example embodiments of the present invention;

5 FIG. 34 illustrates BIM-based robot local navigation with respect to a target nearby building object (lift), according to various example embodiments of the present invention;

FIGs. 35A to 35C show the BIM-based rectification result at the hallway for the ground truth, the coarse localization and the BIM-based rectification, respectively, according to various example embodiments of the present invention;

10 FIGs. 36A to 36D show the PF-based fine tuning result at the hallway for the ground truth, the coarse localization, the BIM-based rectification and the PF-based fine tuning, respectively, according to various example embodiments of the present invention;

FIG. 37 shows Table 4 presenting the robot localization results for the present initialization method in various experiments performed, according to various example embodiments of the present invention;

15 FIG. 38 shows a conventional marker-based localization method including marker deployment and marker detection;

FIG. 39 shows Table 5 presenting the robot localization results for the conventional marker-based localization method in various experiments performed;

20 FIGs. 40A to 40D show a conventional robot localization process with AMCL for randomly distributed particles, multiple particle clouds, confirmed particle cloud and final converged particle cloud, respectively;

FIG. 41 shows Table 6 presenting the robot localization results for the conventional localization method with AMCL in various experiments performed;

25 FIG. 42A shows Table 7 presenting robot localization statistics obtained for various localization methods in various experiments performed;

FIGs. 42B to 42D show plots of experimental evaluations with comparative baseline methods for: moving distance (FIG. 42B), translational localization error (FIG. 42C), and rotational error (FIG. 42D);

30 FIGs. 43A and 43B show localization problems with the conventional robot localization process with AMCL, including wrong localization (FIG. 43A) and inaccurate localization (FIG. 43B);

FIG. 44A shows a BIM model of residential Block A;

FIG. 44B shows robot starting positions at residential Block A in various experiments performed;

FIG. 45A shows a BIM model of residential Block B;

FIG. 45B shows robot starting positions at residential Block B in various experiments performed;

FIGs. 46 and 47 show Tables 8 and 9 presenting robot localization results for residential Blocks A and B, respectively;

FIGs. 48A to 48D show robot multi-level localization results at the residential blocks for the ground truth, the coarse localization, the BIM-based rectification and the PF-based fine tuning, respectively;

FIGs. 49A to 49F show plots of experimental results for different environments: number of detected objects for coarse initialization (FIG. 49A), moving distances for coarse and fine initialization (FIG. 49B), translational localization error in the university building environment (FIG. 49C), translational localization error in residential construction environments (FIG. 49D), rotational localization error in the university building environment (FIG. 49E) and rotational localization error in residential construction environments (FIG. 49F); and

FIG. 50 shows Table 10 presenting robot localization statistics of the experimental results for different environments.

DETAILED DESCRIPTION

[0011] Various embodiments of the present invention provide a method and a system for mobile robot initialization in a building based on a building information model (BIM) of the building, and a mobile robot with automated (or autonomous) mobile robot initialization functionality.

[0012] As explained in the background, conventionally, mobile robot initialization in a building remains a challenge for various working/operating environments, such as in building construction sites where telecommunication infrastructure is not yet available, operating environments are highly repetitive and/or symmetric, and mobile robots are expected to work in changing environments, such as different construction blocks. In this regard, various embodiments of the present invention provide a method of mobile robot initialization in a building, and a system thereof, that seeks to overcome, or at least ameliorate, one or more deficiencies in conventional methods or approaches in mobile robot initialization in a building, and more particularly, to provide automated mobile robot initialization with enhanced or

improved robustness or effectiveness, for example, capable of operating under a variety of working/operating environments, including challenging environments such as GPS-denied environments, highly repetitive and/or symmetric environments and/or changing working environments.

5 **[0013]** FIG. 1 depicts a schematic flow diagram of a method 100 of mobile robot initialization in a building based on a BIM of the building, according to various embodiments of the present invention. The method 100 comprises: obtaining (at 106), from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building; detecting (at 108) a plurality of building objects of the building from
10 the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects; matching (at 110) the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and estimating (at 112) a position of the mobile robot in a BIM frame of the BIM based
15 on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of detected building objects.

[0014] It will be appreciated by a person skilled in the art that the present invention is not limited to any specific type(s) of building objects that may be detected as long as such building objects (including building components/elements) can be detected by a trained building object
20 detector. It will be understood by a person skilled in the art that many different types of building objects (including building components/elements) are possible and are within the scope of the present invention, and it is not necessary to list each and every possible building object herein for clarity and conciseness. By way of examples only and without limitation, various types of building objects of a building include a door, a window, a staircase, a lift and an escalator and
25 so on. In various embodiments, types of building objects that may be detected include those that are included in the BIM of the building.

[0015] In various embodiments, the above-mentioned sequence of detected building objects is in a time order in which the building objects were detected (i.e., chronological order, such as from a first detected building object to a last or latest detected building object in the sequence
30 of detected building objects). In various embodiments, the plurality of building objects detected from the plurality of images by the trained building object detector may be processed or filtered to avoid duplicated detected building objects (i.e., avoid multiple instances of the same building object) in the sequence of detected building objects obtained.

[0016] It will be understood by a person skilled in the art that a BIM of a building is a digital representation of the building, including building objects thereof, and may be designed or created for the building according to techniques known in the art and thus need not be described herein for clarify and conciseness. For example, a BIM may be created for a building being constructed at a building construction site and building objects of the building may be digitally represented as BIM building objects in the BIM. In this regard, it will be appreciated by a person skilled in the art that the BIM includes geometric and semantic information or data of the BIM building objects.

[0017] Accordingly, the method 100 of mobile robot initialization in a building based on a BIM of the building according to various embodiments of the present invention advantageously provides automated (or autonomous) mobile robot initialization with enhanced or improved robustness or effectiveness. In particular, the method 100 advantageously estimates a position of the mobile robot in the BIM frame of the BIM using a matching sequence of BIM building objects obtained based on matching a sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building. In this regard, since the positions of BIM building objects are known in the BIM, determining the position of the mobile robot in the BIM frame by finding matching BIM building objects advantageously enables the mobile robot initialization to operate in challenging environments (e.g., environments with no or insufficient wireless communication/navigation signal coverage (e.g., GPS-denied environments)). Furthermore, by matching a sequence of detected building objects with respect to a plurality of BIM building objects to obtain a matching sequence of BIM building objects, matching BIM building objects for estimating the position of the mobile robot can be found effectively (i.e., improved confidence or accuracy), even in challenging environments (e.g., highly repetitive and/or symmetric environments and/or changing working environments). Therefore, the method 100 of mobile robot initialization in a building based on a BIM of the building according to various embodiments of the present invention advantageously provides automated mobile robot initialization with enhanced or improved robustness or effectiveness, for example, capable of operating under a variety of working/operating environments, including challenging environments such as GPS-denied environments, highly repetitive and/or symmetric environments and/or changing working environments. These advantages or technical effects, and/or other advantages or technical effects, will become more apparent to a person skilled in the art as the method 100 of mobile robot initialization, as well as the corresponding system for mobile robot initialization, is

described in more detail according to various embodiments and example embodiments of the present invention.

[0018] In various embodiments, the above-mentioned matching (at 110) the sequence of detected building objects comprises matching a sequence of groups of detected building objects of the sequence of detected building objects with respect to the plurality of BIM building objects to obtain, for each group of the sequence of groups of detected building objects, one or more matching sub-sequences of BIM building objects for the group of detected building objects based on a matching condition with respect to the group of detected building objects. In this regard, the one or more matching sub-sequences of BIM building objects obtained for the group of detected building objects based on the matching condition is based on object classification information and object position-based information of the group of detected building objects and object classification information and object position-based information of the one or more matching sub-sequences of BIM building objects.

[0019] In various embodiments, the above-mentioned object classification information of the group of detected building objects comprises, for each detected building object of the group of detected building objects, object classification information (e.g., label information) of the detected building object. In various embodiments, the above-mentioned object position-based information of the group of detected building objects comprises, for each pair of a plurality of pairs of detected building objects of the group of detected building objects, relative distance information and relative direction information (e.g., relative direction vector) of the pair of detected building objects. Similarly, the above-mentioned object classification information of the one or more matching sub-sequences of BIM building objects comprises, for each BIM building object of each of the one or more matching sub-sequences of BIM building objects, object classification information of the BIM building object. Similarly, the above-mentioned object position-based information of the one or more matching sub-sequences of BIM building objects comprises, for each pair of a plurality of pairs of BIM building objects of each of the one or more matching sub-sequences of BIM building objects, relative distance information and relative direction information of the pair of BIM building objects.

[0020] In various embodiments, the above-mentioned object position-based information of the group of detected building objects further comprises rotation angle information of the group of detected building objects determined based on the relative direction information of each pair of the plurality of pairs of detected building objects of the group of detected building objects. Similarly, the above-mentioned object position-based information of the one or more matching

sub-sequences of BIM building objects further comprises, for each of the one or more matching sub-sequences of BIM building objects, rotation angle information of the matching sub-sequence of BIM building objects determined based on the relative direction information of each pair of the plurality of pairs of BIM building objects of the matching sub-sequence of BIM building objects.

[0021] In various embodiments, each subsequent group with respect to an immediately preceding group of the sequence of groups of detected building objects partially overlaps with the immediately preceding group of detected building objects. In various embodiments, for the above-mentioned each subsequent group of detected building objects, the one or more matching sub-sequences of BIM building objects obtained for the subsequent group of detected building objects based on the match condition is further based on the one or more matching sub-sequences of BIM building objects obtained for the immediately preceding group of detected building objects.

[0022] In various embodiments, the above-mentioned each subsequent group of detected building objects comprises: an immediately subsequent detected building object of the sequence of detected building objects with respect to a last detected building object of the immediately preceding group of detected building objects, and all detected building objects of the immediately preceding group of detected building objects except a first detected building object thereof. Similarly, for each of the one or more matching sub-sequences of BIM building objects obtained for the subsequent group of detected building objects, the matching sub-sequence of BIM building objects obtained comprises: a BIM building object matching the immediately subsequent detected building object of the subsequent group of detected building objects, and all BIM building objects of a related matching sub-sequence of the one or more matching sub-sequences of BIM building objects obtained for the immediately preceding group of detected building objects except a last BIM building object thereof.

[0023] In various embodiments, the above-mentioned matching (at 110) the sequence of detected building objects further comprises determining the one or more matching sequences of BIM building objects for the sequence of detected building objects, each of the one or more matching sequences of BIM building objects being determined based on a corresponding matching sub-sequence of BIM building objects obtained for a last group of the sequence of groups of detected building objects and based on, for each remaining group of the sequence of groups of detected building objects, a related matching sub-sequence of the one or more matching sub-sequences of BIM building objects obtained for the remaining group of detected

building objects related to the corresponding matching sub-sequence of BIM building objects obtained for the last group of detected building objects.

[0024] In various embodiments, each group of the sequence of groups of detected building objects has three detected building objects (i.e., only has three detected building objects).

5 **[0025]** In various embodiments, the one or more matching sequences of BIM building objects for the sequence of detected building objects comprises a plurality of matching sequences of BIM building objects for the sequence of detected building objects. The above-mentioned estimating (at 112) the position of the mobile robot in the BIM frame is based on a matching sequence of the plurality of matching sequences of BIM building objects that has a
10 least matching error amongst a plurality of matching errors determined for the plurality of matching sequences of BIM building objects, respectively.

[0026] In various embodiments, the method 100 further comprises validating the matching sequence of BIM building objects comprising: selecting a target BIM building object from the plurality of BIM building objects; controlling the mobile robot to navigate to a target location
15 in the building with respect to the target BIM building object; detecting, using the image sensor and the trained building object detector, one or more new building objects in the building as the mobile robot navigates to the target location to obtain an updated sequence of detected building objects including the one or more new building objects; matching the updated sequence of detected building objects with respect to the plurality of BIM building objects to obtain an
20 updated matching sequence of BIM building objects for the updated sequence of detected building objects; and updating the estimated position of the mobile robot in the BIM frame based on the updated matching sequence of BIM building objects for the updated sequence of detected building objects.

[0027] In various embodiments, the above-mentioned matching the updated sequence of
25 detected building objects comprises matching, for each of the one or more new building objects in the updated sequence of detected building objects, a group (e.g., a new group) of detected building objects with respect to the plurality of BIM building objects to obtain one or more matching sub-sequences of BIM building objects for the group (e.g., the new group) of detected building objects based on the matching condition with respect to the group of detected building
30 objects, the group of detected building objects comprising the new building object and a plurality of immediately preceding detected building objects with respect to the new building object in the updated sequence of detected building objects.

[0028] In various embodiments, the target BIM building object is selected from the plurality of BIM building objects based on a distance of the target BIM building object to the position of the mobile robot and a classification of the target BIM building object.

5 [0029] In various embodiments, the method 100 further comprises adjusting the estimated position of the mobile robot in the BIM frame based on determining that the estimated position of the mobile robot is within an inaccessible region (inaccessible by the mobile robot) of the building based on geometric constraint information extracted from the BIM of the building.

10 [0030] In various embodiments, the above-mentioned adjusting the estimated position of the mobile robot in the BIM frame comprises: extracting a nearest BIM wall to the mobile robot at the estimated position from the BIM of the building; and adjusting the estimated position of the mobile robot based on a predetermined spacing to the nearest BIM wall (e.g., a predefined threshold or a minimum allowable distance between the mobile robot to a wall) and a detected distance of the mobile robot to a latest building object detected.

15 [0031] FIG. 2 depicts a schematic block diagram of a system 200 for mobile robot initialization in a building based on a BIM of the building according to various embodiments of the present invention, corresponding to the above-mentioned method 100 of mobile robot initialization in a building as described hereinbefore according with reference to FIG. 1 according to various embodiments of the present invention. The system 200 comprises: at least one memory 202; and at least one processor 204 communicatively coupled to the at least one
20 memory 202 and configured to perform the method 100 of mobile robot initialization in a building as described hereinbefore according to various embodiments of the present invention. Accordingly, the at least one processor 204 is configured to: obtain, from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building; detect a plurality of building objects of the building from the plurality
25 of images obtained using a trained building object detector to obtain a sequence of detected building objects; match the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and estimate a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the
30 one or more matching sequences of BIM building objects for the sequence of detected building objects.

[0032] It will be appreciated by a person skilled in the art that the at least one processor 204 may be configured to perform various functions or operations through set(s) of instructions

(e.g., software modules) executable by the at least one processor 204 to perform various functions or operations. Accordingly, as shown in FIG. 2, the system 200 may comprise: an image obtaining module (or an image obtaining circuit) 206 configured to obtain, from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building; a building object detecting module (or a building object detecting circuit) 208 configured to detect a plurality of building objects of the building from the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects; a building object sequence matching module (or a building object sequence matching circuit) 210 configured to match the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and a position estimating module (or a position estimating circuit) 212 configured to estimate a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of detected building objects.

[0033] It will be appreciated by a person skilled in the art that the above-mentioned modules are not necessarily separate modules, and two or more modules may be realized by or implemented as one functional module (e.g., a circuit or a software program) as desired or as appropriate without deviating from the scope of the present invention. For example, two or more of the image obtaining module 206, the building object detecting module 208, the building object sequence matching module 210 and the position estimating module 212 may be realized (e.g., compiled together) as one executable software program (e.g., software application or simply referred to as an “app”), which for example may be stored in the at least one memory 202 and executable by the at least one processor 204 to perform the corresponding functions or operations as described herein according to various embodiments.

[0034] In various embodiments, the system 200 for mobile robot initialization corresponds to the method 100 of mobile robot initialization as described hereinbefore with reference to FIG. 1, therefore, various operations, functions or steps configured to be performed by the least one processor 204 may correspond to various operations, functions or steps of the method 100 described hereinbefore according to various embodiments, and thus need not be repeated with respect to the system 200 for clarity and conciseness. In other words, various embodiments described herein in context of methods (e.g., the method 100 of mobile robot initialization) are analogously valid for the corresponding systems or devices (e.g., the system 200 for mobile

robot initialization), and vice versa. For example, in various embodiments, the at least one memory 202 may have stored therein the image obtaining module 206, the building object detecting module 208, the building object sequence matching module 210 and/or the position estimating module 212, which respectively correspond to various operations, functions or steps of the method 100 of mobile robot initialization as described hereinbefore according to various embodiments, which are executable by the at least one processor 204 to perform the corresponding operations, functions or steps as described herein.

[0035] A computing system, a controller, a microcontroller or any other system providing a processing capability may be provided according to various embodiments in the present invention. Such a system may be taken to include one or more processors and one or more computer-readable storage mediums. For example, the system 200 described hereinbefore may include at least one processor (or controller) 204 and at least one computer-readable storage medium (or memory) 202 which are for example used in various processing carried out therein as described herein. A memory or computer-readable storage medium used in various embodiments may be a volatile memory, for example a DRAM (Dynamic Random Access Memory) or a non-volatile memory, for example a PROM (Programmable Read Only Memory), an EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM), or a flash memory, e.g., a floating gate memory, a charge trapping memory, an MRAM (Magnetoresistive Random Access Memory) or a PCRAM (Phase Change Random Access Memory).

[0036] In various embodiments, a “circuit” may be understood as any kind of a logic implementing entity, which may be special purpose circuitry or a processor executing software stored in a memory, firmware, or any combination thereof. Thus, in an embodiment, a “circuit” may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g., a microprocessor (e.g., a Complex Instruction Set Computer (CISC) processor or a Reduced Instruction Set Computer (RISC) processor). A “circuit” may also be a processor executing software, e.g., any kind of computer program, e.g., a computer program using a virtual machine code, e.g., Java. Any other kind of implementation of various functions or operations may also be understood as a “circuit” in accordance with various other embodiments. Similarly, a “module” may be a portion of a system according to various embodiments in the present invention and may encompass a “circuit” as above, or may be understood to be any kind of a logic-implementing entity therefrom.

[0037] Some portions of the present disclosure are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a

computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[0038] The present specification also discloses a system (e.g., which may also be embodied as one or more devices or apparatuses), such as the system 200, for performing various operations, functions or steps of various methods described herein. Such a system may be specially constructed for the required purposes or may comprise a general purpose computer system selectively activated or reconfigured by a computer program stored in the computer system. In general, various algorithms that may be presented herein are not limited to being implemented or executed by any particular computer system. Alternatively, the construction of more specialized computer system to perform various operations, functions or steps of various methods described herein may be provided as desired or as appropriate without going beyond the scope of the present invention.

[0039] In addition, the present specification also at least implicitly discloses computer program(s) or software/functional module(s), in that it would be apparent to a person skilled in the art that various operations, functions or steps of various methods described herein may be put into effect by computer code. The computer program(s) is not intended to be limited to any particular programming language and implementation thereof, and it will be appreciated by a person skilled in the art that a variety of programming languages and coding thereof may be used to implement the computer program(s). Moreover, the computer program(s) is not intended to be limited to any particular control flow as there are a variety of programming languages which can use different control flows. It will be appreciated by a person skilled in the art that a computer program may be stored on any computer-readable storage medium (non-transitory computer-readable storage medium), such as but not limited to, a magnetic disk, an optical disk or a memory chip. For example, a computer program stored on a computer-readable storage medium may be loaded and executed on a computer system to implement various operations, functions or steps of various methods described herein according to various embodiments of the present invention.

[0040] Accordingly, in various embodiments, there is provided a computer program product, embodied in one or more computer-readable storage mediums (non-transitory computer-readable storage medium), comprising instructions (e.g., the image obtaining module 206, the building object detecting module 208, the building object sequence matching module 210 and/or the position estimating module 212) executable by one or more computer processors to perform a method 100 of mobile robot initialization as described hereinbefore with reference to FIG. 1 according to various embodiments of the present invention. Accordingly, various computer programs or software modules described herein may be stored in a computer program product receivable by a system therein, such as the system 200 as shown in FIG. 2, for execution by at least one processor 204 of the system 200 to perform various operations, functions or steps of various methods described herein according to various embodiments of the present invention.

[0041] It will be appreciated by a person skilled in the art that various modules described herein (e.g., the image obtaining module 206, the building object detecting module 208, the building object sequence matching module 210 and/or the position estimating module 212) may be software module(s) realized by computer program(s) or set(s) of instructions executable by a computer processor to perform various functions or operations. Various modules described herein (e.g., the image obtaining module 206, the building object detecting module 208, the building object sequence matching module 210 and/or the position estimating module 212) may also be implemented as hardware module(s) being functional hardware unit(s) designed to perform various functions or operations. More particularly, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). Numerous other possibilities exist. It will also be appreciated by a person skilled in the art that a combination of hardware and software modules may be implemented. Furthermore, various operations, functions or steps of various methods described herein may be performed in parallel rather than sequentially as desired or as appropriate (e.g., as long as it does not render the method(s) inoperable or unsatisfactory for its intended purpose).

[0042] FIG. 3 depicts a schematic drawing of a mobile robot 300 with automated mobile robot initialization capability or functionality according to various embodiments of the present invention. In particular, the mobile robot initialization capability or functionality is provided by the method 100 or system 200 for mobile robot initialization as described herein according to various embodiments of the present invention. The mobile robot 300 comprises: a mobile

platform 310 configured to navigate in a building; and a sensor system 320 mounted on the mobile platform 310, the sensor system 320 comprising an image sensor configured to capture a plurality of images of scenes in the building as the mobile robot 300 navigates in the building. The mobile robot 300 further comprises the system 200 as described herein according to various
5 embodiments of the present invention communicatively coupled to the mobile platform 310 and the sensor system 320 (or more particularly, the at least one processor 204 of the system 200 is communicatively coupled to the mobile platform 310 and the sensor system 320) for performing mobile robot initialization in the building based on a BIM of the building.

[0043] As described hereinbefore, the system 200 comprises: at least one memory 202; and
10 at least one processor 204 communicatively coupled to the at least one memory 202 and configured to perform the method 100 of mobile robot initialization in a building as described hereinbefore according to various embodiments of the present invention. Accordingly, in various embodiments, the at least one processor 204 is communicatively coupled to the sensor system 320 to obtain one or more types of sensor data, such as a plurality of images of scenes
15 in the building as the mobile robot 300 navigates in the building. In various embodiments, the at least one processor 204 is communicatively coupled to the mobile platform 310 to control the mobile platform 310 to navigate (e.g., autonomously) in the building, such as to navigate to a target location in the building with respect to the target BIM building object. In various embodiments, the at least one processor 204 is configured to control the mobile platform 310
20 to navigate in the building based on a robot operating system (ROS) (e.g., stored in the at least one memory 202).

[0044] In various embodiments, the sensor system 320 may include a plurality of different types of sensors (perception sensors) for sensing a surrounding environment for generating a plurality of different types of sensor data, respectively. For example and without limitation, the
25 sensor system 320 may include an imaging sensor (e.g., a camera, such as a RGB-D camera configured to provide both depth (D) and colour (RGB) data, which may also have PTZ (pan-tilt-zoom) capability), a distance sensor (a LiDAR sensor) and/or an inertial measurement unit (IMU) sensor.

[0045] It will be understood by a person skilled in the art that the present invention is not
30 limited to any particular type or configuration of mobile robot, as well as any particular type or configuration of mobile platform, as long as the method 100 or the system 200 for mobile robot initialization as described herein according to various embodiments can be implemented in the mobile robot to provide the mobile robot with mobile robot initialization capability or

functionality. In various embodiments, the mobile robot may be ground or aerial mobile robot. Therefore, the mobile platform 310 configured to navigate in a building may be a ground mobile platform or an aerial mobile platform. Various configurations and operating mechanisms or principles of mobile robots (e.g., robot operating system (ROS)), including ground and aerial mobile robots, as well as ground and aerial mobile platforms, are known in the art and thus need not be described herein for clarity and conciseness.

[0046] It will be appreciated by a person skilled in the art that the terminology used herein is for the purpose of describing various embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0047] Any reference to an element or a feature herein using a designation such as “first”, “second” and so forth does not limit the quantity or order of such elements or features, unless stated or the context requires otherwise. For example, such designations may be used herein as a convenient way of distinguishing between two or more elements or instances of an element. Thus, a reference to first and second elements does not necessarily mean that only two elements can be employed, or that the first element must precede the second element, unless stated or the context requires otherwise. In addition, a phrase referring to “at least one of” a list of items refers to any single item therein or any combination of two or more items therein.

[0048] In order that the present invention may be readily understood and put into practical effect, various example embodiments of the present invention will be described hereinafter by way of examples only and not limitations. It will be appreciated by a person skilled in the art that the present invention may, however, be embodied in various different forms or configurations and should not be construed as limited to the example embodiments set forth hereinafter. Rather, these example embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the present invention to those skilled in the art.

[0049] In particular, for better understanding of the present invention and without limitation or loss of generality, various example embodiments of the present invention will now be described with respect to methods and systems for mobile robot initialization whereby the

mobile robot is a ground mobile robot and the building is a building being constructed at a building construction site for illustration purposes only. It will be understood by a person skilled in the art that the present invention is not limited to a ground mobile robot and not limited to being implemented in, or applied to, a building construction site. For example, the building may be any building in which a mobile robot may be deployed for navigating therein to perform various tasks and the mobile robot may be an aerial mobile robot without going beyond the scope of the present invention.

[0050] Conventionally, for most applications of construction mobile robots, the charging station is taken as a fixed starting point with known coordinates, and the mobile robot always starts navigating from that fixed starting position. Besides the conventional approach of using a fixed starting position, artificial markers can be deployed to initialize the mobile robot by decoding the embedded location information. However, such arrangements limit the flexibility of the mobile robot system and require a manual process that always takes the mobile robot to a particular location to initiate the navigation, or to deploy markers throughout the working environments, which are time-consuming and inconvenient. In addition, the mobile robot can lose its location information due to the accumulated error in path tracking, or because its operation is interrupted and it is moved to other locations. To resume operation and navigating, the identification of the instant/current location of the mobile robot in the map is required to re-initialize the mobile robot, and the tedious manual initialization is required for each position loss based on the conventional method of using a fixed starting position for example.

[0051] Accordingly, positioning and localization for indoor mobile applications is a fundamental function to carry out location-based services, such as indoor navigation, time-critical rescue, and all robot-related applications. Radio-frequency (RF) based methods using dedicated wireless communication networks were extensively studied. However, they were infrastructure-dependent due to the necessity for the deployment of signal transmitters and beacons. With the advances and ubiquitous availability of inertial sensors, cameras, and LiDAR, infrastructure-free methods using visual markers and simultaneous localization and mapping (SLAM) technology were developed. However, pre-scanning and pre-configuration for each different environment was usually performed with these conventional methods, making them time-consuming and inefficient, especially when applied in changing environments such as in building construction sites. BIM that contained substantial information about the environments have also been explored to integrate with the existing localization methods to either improve the localization accuracy or reduce the effort for environment configuration in construction

automation. For better understanding, various conventional infrastructure-dependent methods, various conventional infrastructure-free methods and various conventional BIM-based localization will now be described below.

[0052] In relation to conventional infrastructure-dependent methods, RF-based methods estimated the position by using the mobile device and the deployed infrastructure as a signal transmitter and receiver pair. The location was determined based on the measured distances and angles to the infrastructure with the known locations or the measured signal strength pattern. A wireless fidelity (WiFi)-based indoor positioning system was developed for labour tracking at a shield tunnel construction site. Various other conventional methods used ultra-wideband (UWB) beacons and radio frequency identification (RFID) tags to integrate with robot motion data for positioning. However, these RF-based methods are not suitable for environments with no or insufficient wireless communication/navigation signal coverage such as in construction environments since communication devices are not yet available and the deployment of infrastructure is time-consuming and not cost-effective. In addition, the movement of the workers and construction materials can cause instability in received signal strength and introduce severe multipath problems that impair localization accuracy.

[0053] In relation to conventional infrastructure-free methods, various infrastructure-free localization methods were developed using visual and laser sensor data to eliminate the requirement for deployment of signal transmitters and beacons. Fiducial markers, real visual features in images, and laser scans were used to recognize locations through feature matching and scene understanding.

[0054] For example, one of the techniques for accurate and economical localization was to deploy artificial markers such as QR codes and ArUco tags with encoded location information (i.e., marked-based methods). QR codes were used as global reference points for a mobile robot localization system with the markers deployed on the ceilings. A more generalized and real-time QR code-based method was proposed in to estimate the 3D pose of the camera with varying orientations based on extended image processing. However, marker-based methods still require substantial effort to configure the working environment. When a robot is deployed to detect markers through self-exploration, markers can be easily missed or occluded by obstacles. Moreover, the markers are susceptible to wear and tear during the construction work, and thus result in localization failure.

[0055] For example, in indoor GPS-challenged environments, simultaneous localization and mapping (SLAM) technology (i.e., SLAM-based methods) is usually used for mobile

robots to generate a map of the unknown environment while localizing the robots in the constructed map without any infrastructure deployment. Visual SLAM (vSLAM) was extensively studied for cost-effective indoor localization. Mono-SLAM estimated the pose state through matching of image patches and optimized the estimation using extended Kalman filter (EKF). There was also introduced the concept of keyframes and proposed a framework of parallel tracking and mapping (PTAM). Improved from PTAM, another keyframe-based system, ORB-SLAM, was developed for tracking, mapping, relocalization, and loop closing, and obtained high popularity in commercialized applications. This was extended to ORB-SLAM2 for RGB-D cameras. OpenVSLAM was proposed based on ORB-SLAM while improved for higher extensibility and accuracy. In the stack of vSLAM, relocalization was an essential module to determine the camera pose from the built map and recover from tracking failure using place recognition techniques. In construction automation, instead of the location in the vSLAM map, the robot's initial position in a reference coordinate system was expected to navigate to a target location with known positions in the same coordinate system. There was also disclosed an in-building emergency response system which marked the fixed target positions in the built feature map from vSLAM to link the initial position and destinations. However, accurate mapping using vSLAM with exact matching to the physical world was challenging for dynamic construction environments with unstructured regions, temporarily placed materials, and presence of workers. There were also disclosed techniques whereby the geo-tagged database of images or features was constructed to directly match the query image to the reference image with the known absolute location in a global coordinate system using visual features for localization. The recent success of deep neural networks on a variety of domains has boosted research on vision-based localization using convolutional neural network (CNN) for more robust image retrieval. For example, Mask R-CNN network was used to detect objects in the image, and SURF descriptors were applied to each object to retrieve object information for keyframe matching. As another example, VGG16 based image descriptors were applied to extract Top-N similar images, and Faster RCNN based region descriptors were leveraged to re-rank the image candidates for robust image retrieval. Although these previous works achieved infrastructure-free localization, it required time-consuming pre-mapping of the whole working environment to build a high-resolution geo-tagged database of visual features for accurate localization. Moreover, the rapid variations of scenes due to ongoing construction work can easily cause failure in detecting corresponding keyframes and a frequent construction of the visual feature database to overcome such a problem is inefficient.

[0056] Besides vSLAM, laser-based SLAM algorithms with higher accuracy and faster computation were also explored for robot localization. Gmapping was a widely used SLAM algorithm based on the improved Rao-Blackwellized Particle Filter (RBPF). Another efficient 2D SLAM algorithm, Hector SLAM, used a bilinear filtering function to estimate grid occupancy and Gauss Newton method to optimize scan matching for a real-time localization. However, these Bayes-based methods lacked loop closure detection and optimization and therefore, were not capable of eliminating accumulated localization errors. The graph-based SLAM, such as Karto SLAM, and Cartographer, in contrast, used observations as constraints and modeled a graph-based representation for global optimization which minimized accumulated drift errors. The Cartographer algorithm that included local scan matching, back-end loop closure detection, and subgraph optimization achieved high accuracy and robust real-time performance. The location tracking methods in laser-based SLAM could also be used to globally localize the robot from an unknown initial position in a given map. Scan matching algorithms were applied to match the query scan to a database of keypoints generated from the grid-based maps and retrieve the associated location information. There were also disclosed other probabilistic localization algorithms that are variants of the Bayes filter which transformed the probabilistic belief of robot pose with robot motion and observation models. However, the prior mapping of the entire environment was required with these conventional methods, and the discrepancies between the built map and the physical environment resulted in inaccurate global localization. The AMCL software offered a convenient registration between a 2D ground truth map, such as a designed floor plan of a building, to the robot laser scans for fine-grained global localization. Although it performed well for environments with rich geometric features, it was vulnerable to wrong localization in repetitive environments. Therefore, it was not feasible for large-scale complex construction environments with many similar plain walls, pillars and long corridors.

[0057] In relation to conventional BIM-based localization, a number of researches have studied the potential of BIM in robot path planning and position tracking using the indoor mobility information in BIM, such as the size of areas and the accessibility to other spaces and transitions. BIM was also used in infrastructure-dependent indoor localization systems to provide the environment information. A hybrid asset tracking system for location awareness in indoor construction environments was developed integrating the Bluetooth Low Energy (BLE) technology, motion sensors and BIM. A BIM-based indoor localization algorithm was also proposed to assist the building fire emergency response operations. Algorithms for beacon

deployment leveraging BIM were developed and the proposed localization schema helped improve the accuracy of room-level localization. However, the above-mentioned conventional methods still relied on traditional RF-based localization approaches and hence, required device deployment.

5 [0058] Various other conventional methods may fuse BIM information with visual and LiDAR observations to eliminate online surveying operation in traditional SLAM-based methods. For example, VGG network was implemented to extract image features and match captured real images with a configured geo-tagged database of rendered BIM images. However, it only determined the matched image while the exact camera pose was not estimated. There
10 was also developed a camera pose regressor based on the PoseNet architecture with synthetic training images rendered from a 3D BIM model. However, the error of 2 meters for localization with this method was insufficient for accurate robot navigation. Although the time taken for on-site image collection was reduced in the above-mentioned conventional methods, additional offline effort to construct a geo-tagged dataset for a large number of rendered BIM images was
15 still time-consuming. As the layouts varied for different blocks, a separate database of BIM images is required for each different environment, making it environment-dependent and less efficient. There was also disclosed a BIM-based interface to convert a BIM model into serialized state representations of a 3D SLAM. It localized the robot by matching the online SLAM with the composed mapping state. This method could be used to recognize the robot's
20 location without surveying the entire working space. However, it still required the robot to start from a pre-mapped zone where the region around the initial position should be sufficiently similar to the reference model. Therefore, it is not suitable for highly repetitive and dynamic environments, such as construction environments with rooms of the same/similar size, long corridors without closed boundaries in the laser-scans, and randomly placed construction
25 materials that introduced uncertainty in scan matching. Such characteristics may likely result in poor initialization due to the lack of a unique matched scan.

[0059] In contrast, various example embodiments provide a BIM-based mobile robot initialization system which is automated (or autonomous), infrastructure-free and environment-independent. As an example practical application, the BIM-based mobile robot initialization
30 system may be applied to facilitate or enable fully automated robot navigation for construction automation without any pre-scanning of the environment, such as autonomously localize the mobile robot in a known map built from BIM, according to various example embodiments of the present invention. In various example embodiments, CNN-based object detection is used to

recognize building objects (e.g., visual features) while geometric and semantic BIM information are utilized to associate with the sensory information to facilitate the mobile robot location estimation. Accordingly, for example, automated robot initialization in various construction environments can advantageously be realized without any requirement for pre-scanning or infrastructure deployment.

[0060] Accordingly, various example embodiments provide a BIM-based feature (building object) association approach and CNN-based real-time object detection to realize the mobile robot self-initialization. The method of mobile robot initialization is infrastructure-free and therefore, can eliminate human effort for mobile robot initialization. The method is also environment-independent and requires no prior mapping or scanning of the entire environment when applied to new environments with different scenes and layouts. For example, the BIM information retrieval may be standardized (e.g., for all BIM files) while the same object detector is employable for different environments that exhibit similar visual features, such as different blocks at the same construction site. For example, the environment-independent characteristic can improve the mobile robot initialization efficiency at different blocks with various layouts, and can also be generalized easily across different construction sites. In addition, the method yields good performance in adapting to construction environments that are repetitive and symmetric (e.g., plain walls and pillars). Therefore, the method improves the mobile robot initialization efficiency when the mobile robot is required to work in multiple environments. Various example embodiments demonstrate the superiority of using BIM in mobile robot initialization in various aspects. In various example embodiments, the semantic and geometric BIM information is not only used for the feature matching (building object matching) and location estimation, but also utilized for the robot decision making and providing geometric constraints for localization. Therefore, various example embodiments leverage BIM to expedite the initialization process and improve the localization accuracy. Furthermore, in various example embodiments, the online robot decision-making and active exploration techniques are provided by fusing BIM information and sensory data, making the initialization automatic and require no manual control on the mobile robot, and hence contribute towards the fully automated robot navigation system. In various example embodiments, the method further includes multi-level rectification of the mobile robot's final pose for further enhancing accurate initialization results for mobile robot navigation. Accordingly, besides the advantages of relaxing the effort on infrastructure deployment and environment configuration, as well as good performance in complex construction environments, the method of mobile robot initialization

according to various example embodiments provides an integrated and autonomous system that includes techniques/algorithms for overall online decision making, feature (building object) matching, local navigation, and positioning rectification. The method is able to achieve accurate initialization without human intervention, and exhibits the capability to further integrate with the robot navigation system to realize fully autonomous robot navigation, given the mobile robot deployed at an unknown location in a building.

[0061] In various example embodiments, the BIM-based mobile robot initialization system uses a trained building object detector (e.g., a Convolutional Neural Network (CNN)-based object detector) for detecting various building objects as the mobile robot (which may simply referred to as a robot herein) navigates in a building. The system may involve two stages, namely, an offline BIM information retrieval, data collection and training stage (which may simply be referred to as the offline stage) and an online robot initialization stage (which may simply be referred to as the online stage). In the offline stage, geometric and semantic BIM information for building objects (including building elements or components) may be retrieved and an object detector is trained to detect the corresponding building objects. In the online phase, the robot may be deployed at an unknown location in a working/operating environment and may start to determine or estimate its position in the BIM frame automatically based on prior known BIM information (e.g., semantic and geometric information of BIM building objects) and online sensory information (e.g., semantic and geometric information of detected building objects). In various example embodiments, the robot may be configured to explore the surrounding environment to find/detect building objects (e.g., architectural features, such as doors, lifts and staircases and so on) using the building object detector trained in the offline stage. In various example embodiments, each selected building object (including building component/element) may be considered as a separate feature. In this regard, the association between the detected features (detected building objects) and features (BIM building objects) in the BIM is determined based on the relative positions among detected features and geometric information of features in BIM. Thereafter, according to various example embodiments, coarse robot positioning and fine-tuned localization may performed based on the coordinates of detected building objects in the BIM frame and relative positions between the robot and detected building objects.

[0062] FIG. 4 depicts a schematic drawing showing an overview of a robot initialization system 400 (e.g., an integrated robot initialization system), according to various example embodiments of the present invention (e.g., corresponding to the system 200 for mobile robot

initialization as described hereinbefore with reference to FIG. 2 according to various embodiments).

5 [0063] In various example embodiments, an offline image database may be built and utilized to train an object detector for detecting building objects (including building components). Both semantic and geometric information of the building objects may be retrieved from the BIM to provide knowledge of the working/operating environment. In the online stage, the robot may utilize BIM information and online sensory data to execute its motion and estimate its location while exploring the environment.

10 [0064] In various example embodiments, the online initialization stage involves a robot navigation system 406, a vision system 408 (e.g., corresponding to the building object detecting module 208 as described hereinbefore according to various embodiments) and the feature (or building object) matching module 410 (e.g., corresponding to the building object sequence matching module 210 as described hereinbefore according to various embodiments). The vision system 408 may perform object detection, process detection results and send signals to the navigation system 406 for decision making of robot actions, such as stopping of the robot, and recording of feature information (detected building object) once a new feature (building object) is detected by the vision system 408. The feature matching module 410 may obtain BIM information (e.g., semantic and geometric information of BIM building objects) and online sensory information (e.g., semantic and geometric information of detected building objects) to determine the association between detected features (detected building objects) and the BIM feature list (BIM building objects). In various example embodiments, the feature matching module 410 may evaluate the confidence level in which the robot is located at the estimated location based on the matching results.

25 [0065] In various example embodiments, the navigation system 406 is autonomous and may receive signals from the vision system 408 and feature matching module 410 to control the robot's motion. In various example embodiments, the navigation system 406 may be configured with a global exploration mode and a local navigation mode. In the global exploration mode, the robot may explore the environment randomly or without a pre-planned path or target destination and search for features (building objects). In the local navigation mode, a goal or target point or location is provided and the robot motion is planned to reach the target location. 30 The robot may perform global exploration by default and stops once the vision system 408 decides to localize a detected building object. In various example embodiments, after a number of features (e.g., at least three features/building objects) have been detected and the robot

position can be estimated, validation of a matching sequence of features (building objects) may be performed to accelerate the convergence of the feature matching and confirm the robot location. In this regard, the feature matching module 410 may send a signal to the navigation system 406 to execute local navigation to navigate to possible nearby features to build up the confidence level for the estimated robot position.

[0066] For better understanding, the offline and online stages will now be described in further details according to various example embodiments of the present invention. In this regard, it will be appreciated by a person skilled in the art that various example specific implementation or configuration details will be described according to various example embodiments of the present invention, and as explained above, the present invention is not limited to the example embodiments described, let alone the example specific implementation or configuration details.

Offline Phase

[0067] In the offline stage/phase, for example, Autodesk BIM software (Revit) and its plug-in software (Dynamo) may be used to process an industry foundation classes (IFC) data model and retrieve the geometric and semantic information of building objects (e.g., architectural building components). For example, CNN-based improved version of You Only Look Once (YOLOv3) architecture may be used to train the object detector of building objects (including building components/elements) based on a dataset comprising images extracted from various sources, such as internet resources and those captured at university buildings and building construction sites. For example, since real-time object detection is required to carry out the online initialization, YOLOv3 may be chosen for its good performance in real-time applications and high detection accuracy as well. However, the present invention is not limited to the YOLOv3 architecture and other object detection frameworks may instead be used as desired or as appropriate, such as YOLOv4, YOLOR or Mobilenet-SSDv2.

BIM information retrieval

[0068] Various BIM information required for robot initialization and an example approach adopted to acquire information of building objects in a 3D BIM model will now be described according to various example embodiments of the present invention. It will be understood by a person skilled in the art that BIM is an integrated system that allows architecture, engineering, and construction (AEC) professionals to manage the entire lifecycle of building construction.

For example, BIM utilizes an object-based parametric modeling technique and provides a 3D digit visualization of the physical and utilitarian characteristics of a place. As an example illustration, a sample BIM (digital representation) for one single floor in a residential building is shown in FIG. 5. BIM can be treated as a database that contains all the architectural and structural objects (including components/elements), as well as mechanical and electrical services and their corresponding properties. For instance, given a door element (a building object) modeled in BIM, the door dimensions, locations, materials, cost, and other properties associated with the door may be included in the BIM. As an example illustration, an example of a door element and information of the door element that may be visualized in a BIM software is shown in FIGs. 6A to 6D. In particular, FIG. 6A depicts a 3D model of the example door element digitally represented in the BIM, and FIG. 6B, FIG. 6C and FIG. 6D show example geometric information, example location information and example semantic information of the door element, respectively.

[0069] A BIM can be modeled with various software in different file formats. To transfer the model across various hardware devices and software platforms, the Industry Foundation Classes (IFC) was developed as a mainstream BIM standard. IFC is an international standard (ISO 16739-1:2018) that provides a standardized and digital representation of building objects (including building components/elements) and facilitates vendor-neutral interoperability in different applications. BIM data may be retrieved and utilized for different use cases using the IFC data format specification.

[0070] For example, in the robot initialization system according to various example embodiments, the BIM may be converted to the IFC file format and then exported into the Autodesk Revit software, which is a strong development platform to create and manage BIM for architects, landscape architects, structural engineers, mechanical, electrical, and plumbing (MEP) engineers, designers and contractors. Then information retrieval approach may be developed based on the built-in Dynamo software that is developed for graphical programming in Revit. Furthermore, in various example embodiments, an application is created to extract information from BIM and export the extracted information to a text file. For example, a user may select building objects by selecting them from a graphical user interface of BIM software or by specifying a category of interest (e.g., doors, electrical equipment, and so on). As an illustrative example, FIG. 7A depicts an example user interface for BIM information retrieval.

[0071] The selected building objects may then be passed to a customized program configured to retrieve or extract the semantic and geometric attributes of the building objects.

In this regard, an example of the extended graphical programming interface in Dynamo is shown in FIG. 7B. Various example embodiments note that, since the attributes of each category are different, some properties cannot be extracted directly. For example, for a door element, the surface norm to the wall plane where the door is embedded is available. On the other hand, the surface norm of the wall elements is not directly available since a wall is modeled based on a line representing the wall laying direction. The surface norm of the wall is determined using the directional vector to calculate the corresponding perpendicular vector. In various example embodiments, such processing work is performed on all the BIM categories to extract object attributes for each building object. For example, semantic information of a building object may include a unique element ID number and a category (which may also be referred to as classification or label) of the building object. For example, geometric information of a building object may include the location of the building object in the 2D floor plan (in the BIM frame), the normal vector to the front surface of the building object and the dimensions of the building object. By way of examples only and without limitation, FIG. 8 shows Table 1 including example BIM information of various building objects.

[0072] For example, the BIM information list comprising a plurality of BIM building objects may be used to provide the category information of the plurality of BIM building objects and to determine the relative positions among different BIM building objects. In this regard, various example embodiments predict or estimate the robot path or position through matching detected visual features (i.e., detected building objects) with BIM building objects extracted from the BIM.

CNN-based object detector training and interpretation of detection results

[0073] You Only Look Once (YOLO) architecture is a CNN-based object detector that processes an image in a single-stage deep neural network to locate objects of interest in the image. It performs object detection as a regression problem that predicts bounding boxes of objects and associated possibilities of object classes. Among the YOLO family, YOLOv3 is one of the state-of-the-art detectors with a 53-layer CNN-based feature extractor. As an example, the object detector in the robot initialization system according to various example embodiments of the present invention may be implemented based on the YOLOv3 architecture. For example, YOLOv3 exploits the residual connections and utilizes the feature pyramid networks to detect objects in three different scales. Given an input image fed into the framework, the 53 layers CNN-based feature extractor network (Darknet-53) is applied to produce the feature map at one

scale. Then the feature map is upsampled and concatenated with the feature map from previous layers into a combined feature map that is in a new scale. Such a process merges the high-level meaningful semantic information with the finer-grained feature information from previous layers. The same approach is applied again on the second scale map and produces the final scale feature map. For each of the three scales, the object detector predicts a 3-D tensor encoding bounding box, objectiveness score, and class label. Each scale corresponds to a feature map with $S \times S$ grids where 3 bounding boxes are predicted for each grid cell and 4 coordinates t_x , t_y , t_w , t_h are determined for each bounding box. Given c_x , c_y as the offset to the grid cell from the top left corner of the image and p_w , p_h as the width and height of the anchor box, the bounding boxes coordinates, b_x , b_y , b_w , b_h are interpreted using the predicted values. b_x , b_y , b_w , b_h represent the x and y coordinates, width and height of the bounding box, respectively. In this regard, FIG. 9 shows example bounding box predictions with prior anchor boxes. Therefore, for each scale, the dimensions of the predicted tensor are $S \times S \times [3 \times (4 + 1 + C)]$ where C represents the total number of classes. Thereafter, the predicted bounding boxes from three scales may be processed with Non-Maximal Suppression (NMS) to eliminate the multiple overlapped predictions.

[0074] Based on the YOLOv3 framework, the detection model (building object detector) may be trained to recognize various building objects, such as common architectural building objects (including components/elements) at buildings, such as at building construction sites, for example, doors, lifts and staircases. For example, the convolutional weights that are pre-trained on ImageNet using Darknet-53 may be set as the initial weights. For example, training images may be collected from various sources, such as internet resources, ongoing building construction sites and university buildings. Images taken at building construction sites exhibit diversity in lighting conditions, viewing angles and building construction stages to ensure that the object detector trained can be applied at varying construction stages and different building construction sites. By way of examples only, FIG. 10A shows example images collected from different sources for the image dataset for the object detector training.

[0075] For example, training images may be fed into the detector framework with a size of 64 images per batch and weight parameters may be updated after each iteration of processing 16 images. As an example implementation, a total of 4000 iterations were performed to obtain a final detection model. The trained model (trained building object detector) may then be used to detect and localize building objects in a building. By way of examples only, FIG. 10B shows example building object detection results.

Online Phase

[0076] In various example embodiments, the trained real-time object detector and the retrieved BIM information in the offline phase may then be utilized in the online phase/stage.

- 5 FIG. 11 depicts a schematic flow diagram of a method 1100 of mobile robot initialization in a building based on a BIM of the building, according to various example embodiments of the present invention (e.g., corresponding to the method 100 of mobile robot initialization as described hereinbefore according to various embodiments).

[0077] The method 1100 comprises: obtaining (at 1106), from an image sensor mounted on
10 a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building; detecting (at 1108) a plurality of building objects of the building from the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects; matching (at 1110) the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or
15 more matching sequences of BIM building objects for the sequence of detected building objects; and estimating (at 1112) a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of detected building objects.

[0078] In various example embodiments, as the initialization starts from an unknown
20 location in the operating environment, without any knowledge of the surrounding environment, the robot may conduct random exploration (at 1106) to search for visual features (building objects) using the trained object detector. Wall following algorithm may be applied to facilitate the random exploration (without a pre-planned path or target destination). For example, by default, the robot follows the left wall and keeps going straight. It turns right at an inner corner
25 when there is an obstacle in the front and it turns left to find another piece of wall as it reaches an outer corner when the wall it has followed comes to an end. As the robot explores the environment, the real-time object detector may be continuously executed (e.g., at a frequency of 5 images per second) and all the detected building objects are recorded in a memory of the robot. Thereafter, the BIM information (BIM building objects extracted from the BIM) is
30 leveraged to associate with information of the detected building objects and the position of the robot may then be estimated based on the association of building objects (i.e., building object matching). If there are multiple hypotheses (i.e., multiple estimates) of the robot position, in various example embodiments, local navigation (at 1116) using BIM information is triggered

to control the robot to find additional nearby feature(s) (building object(s)) and validate the most probable hypothesis (estimate). In this manner, as more building objects are detected through random exploration and active local navigation, the false hypotheses (estimates) are opted out and the unique match between BIM and the online sensory information (i.e., between detected building objects and BIM building objects) can be determined. This may then complete a first (e.g., coarse) initialization by identifying a first or preliminary (e.g., rough) region where the robot is located. In various example embodiments, multi-level positioning rectification (at 1118) may be applied to fine tune the first (e.g., coarse) initialization result so as to improve the localization accuracy. For better understanding, the method 1100 of mobile robot initialization will now be described in further details according to various example embodiments of the present invention.

Detection and localization of building objects

[0079] The detection and localization of building objects (at 1108) will now be described in further detail according to various example embodiments of the present invention, which corresponds to the image obtaining module 206 and the building object detecting module 208 as described hereinbefore according to various embodiments. In various example embodiments, the detection and localization of building objects may be performed using the pre-trained object detector (e.g., the YOLOv3 object detector) and depth measurements with a RGBD sensor. Moreover, techniques for processing the object detection results and recognize distinct building objects from a continuous image stream will also be described below according to various example embodiments of the present invention. In this manner, information of all building objects that have been detected since the initialization started can be obtained and recorded in a detected feature list (detected building object list).

[0080] As the robot explores and navigates in the environment, the real-time object detector is executed continuously. In each raw image, if a building object (which may simply be referred to as an object) is detected, the 3D location of the object with respect to a fixed odometry frame is calculated. The odometry coordinate frame is established depending on the starting robot pose where the initialization began. The origin of the odometry frame is the starting robot position and the positive x -direction of the odometry frame is the robot facing direction.

[0081] To localize an object from an image captured, camera intrinsic parameters, 2D object location in the image frame and depth to the object are required. Camera calibration is performed, and intrinsic parameters including image center point (p_x, p_y), focal length (f), and

distortion coefficients are obtained. From the object detection results, the object center (b_x, b_y) in the image frame can be obtained. The mean value of the depth measurements acquired from a RGBD sensor in a $(2n + 1) \times (2n + 1)$ pixels square region around the object center, denoted as D , is used to localize the object and therefore, this average depth D may be calculated as:

$$D = \frac{1}{(2n + 1) \times (2n + 1)} \sum_{i=b_x-n}^{b_x+n} \sum_{j=b_y-n}^{b_y+n} d_{ij} \quad (\text{Equation 1})$$

where d_{ij} is the depth measurement at the pixel (i, j) in the image frame.

[0082] Given the centers of the detected object are b_x and b_y in the image frame from object detection results, the object position in the 2D image frame may be transformed to the 3D point in the camera frame as follows:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} (b_x - p_x) \times \frac{D}{f} \\ (b_y - p_y) \times \frac{D}{f} \\ D \end{pmatrix} \quad (\text{Equation 2})$$

where x_c, y_c, z_c represent the coordinates of the detected object with respect to the camera frame.

[0083] The object position in the camera frame may further be converted to the position in the odometry frame through successive homogeneous transformations as follows:

$$\begin{pmatrix} x_o \\ y_o \\ z_o \\ 1 \end{pmatrix} = T_{odom}^{robot} \cdot T_{robot}^{lidar} \cdot T_{lidar}^{camera} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} \quad (\text{Equation 3})$$

where $T_{f_2}^{f_1}$ is the homogenous transformation matrix from one frame f_1 to the other frame f_2 and x_o, y_o, z_o are the coordinates of the detected object in the odometry frame.

[0084] Given the above-mentioned equations, each detected object can be localized to a 3D position in the odometry frame. As an example illustration, FIG. 12 shows the relationships or transformation between different frames, including the BIM frame, the odometry frame, the robot frame, the Lidar frame, the camera frame and the image frame.

[0085] For each raw image, a set which includes the detected objects in the current image and the information of each object, including the class and the position of the object in the odometry frame may be obtained, and denoted as $S_{current}$ (e.g., may be referred to as current image detected object set). This set only includes the information of the objects detected in the current image and hence refreshed as a new image is captured. The information in $S_{current}$ from individual raw images may be aggregated and processed to acquire the information of all the observed objects since the initialization started using a detection filter. The set of all the objects that have been detected (e.g., may be referred to as a current sequence detected object set) until the current image, S_{object} with a total of n^S objects is denoted as:

$$S_{object} = (s_1, s_2, s_3 \dots s_n^s) \quad (\text{Equation 4})$$

where s_i represents the i^{th} observed object.

[0086] The class and the object location for each object are recorded in S_{object} . However, various example embodiments note that the objects in $S_{current}$ cannot be simply directly added to S_{object} for a number of reasons. Firstly, the same objects usually appear in multiple consecutive images. In this regard, in various example embodiments, a filter (the above-mentioned detection filter) is provided to distinguish different objects and group the detections for the same object to avoid duplicated objects. In addition, since the detection is performed as the robot is moving, the vibration of the sensors may cause image blurring and degrade detection accuracy. Random false detections due to instability are possible in some images and in various example embodiments, such unstable detections are discarded by stopping the robot to confirm and localize the object without any robot motion. To address the above-mentioned problems, in various example embodiments, three requirements or conditions are set or predefined in order to accept a detected object as a confirmed detected object for addition to S_{object} , as follows: (1) the object is detected frequently within a few frames, (2) the object can be detected as the robot stops, and (3) the object is not near to any other confirmed objects detected before.

[0087] FIG. 13 depicts a schematic flow diagram of a method 1300 of filtering detected building objects according to various example embodiments of the present invention, and more particularly, based on the above-mentioned three conditions. Accordingly, various example embodiments may provide a detection filter configured to perform the method 1300 of filtering detected building objects.

[0088] As shown in FIG. 13, while there is an object detected in the current image, object information is stored in $S_{current}$. An intermediate set of features (detected objects) $S_{windows}$ is introduced to integrate the information in the past sequence of images, and serves as a short-term memory of the recently detected objects and the frequency of appearance for each object.

5 $S_{windows}$ is defined as a set which includes the objects that appeared in the past $n_{windows}$ images, where $n_{windows}$ is the window size and defined by a user. The locations of objects in $S_{current}$ are compared with those in $S_{windows}$. If the detected object is not close to any object in $S_{windows}$, meaning this object is not detected in the past few seconds, it is added to $S_{windows}$ as a new object. On the other hand, if the distance is smaller than a threshold d_{object} , which

10 indicates the same object has been seen before, the count of occurrence of the object in $S_{windows}$, denoted by n_o is increased by one. Meanwhile, the total number of past images since an object is detected for the first time are counted as m_o . If n_o reaches a certain number n_{check} , indicating that an object is detected several times in the past few seconds and there is a high probability that an object exists, criteria 1 is fulfilled. If so, the vision system may send a signal to the robot

15 navigation system to stop the robot and perform a more accurate detection and localization of the object. In the scenario that m_o is raised to the window size, showing that an object is detected occasionally in the past few frames due to random false detections, the object is discarded from $S_{windows}$ and the robot is not commanded to stop and examine the presence of that object. For the object to be checked in $S_{windows}$, the robot stops and waits for a number of

20 frames (e.g., 15 frames) to be captured before recording the object position to avoid unstable measurement of depth to the object. If the object is present when the robot stops to observe it, criteria 2 is satisfied. The location of the object is compared with those of other confirmed objects in S_{object} . If the object is close to any existing object in S_{object} , showing that the object has been detected and recorded before, the object information is not saved repeatedly. On the

25 contrary, if the distances to all the other objects are larger than d_{object} , the third or last criteria (criteria 3) is satisfied and the object is determined as unique (i.e., not a duplicate) and added to the current sequence detected object set S_{object} .

Building object matching

30 [0089] The matching of detected objects with BIM detected objects (at 1110) will now be described in further details according to various example embodiments of the present invention, which corresponds to the building object sequence matching module 210 as described hereinbefore according to various embodiments.

[0090] As described hereinbefore, a set of objects (S_{object}) detected since the robot started as defined in Equation (4) may be obtained through robot sensing. Similarly, a set of BIM objects (B_{object}) retrieved or extracted from the BIM in the offline phase, with a total of n^B objects may be denoted as:

$$B_{object} = (b_1, b_2, b_3 \dots b_{n^B}) \quad (\text{Equation 5})$$

[0091] For each detected object in S_{object} , there exists an identical or corresponding object in the BIM that matches with it but the detected objects and the BIM objects are defined in different coordinates since the locations of objects in S_{object} and in B_{object} are defined with respect to the odometry frame and BIM frame, respectively. To obtain the locations of detected objects in the BIM frame, the association between these two sets of objects is established according to various example embodiments of the present invention. In this regard, an object matching technique is developed to map each detected object in S_{object} to a corresponding object in BIM in B_{object} using semantic information and geometric relationships between detected objects and BIM objects.

[0092] To illustrate the associations between objects in the two sets of objects (S_{object} and B_{object}) and the relationships between the locations of objects in the odometry frame and BIM frame, an example is shown in FIG. 14. The robot starts at an unknown location and the origin of the odometry frame is this starting point. Suppose that the sequence of the objects detected is (s_1, s_2, s_3, s_4) and there are six BIM objects in the BIM list, $(b_1, b_2, b_3, b_4, b_5, b_6)$, a feasible match for this example is denoted as $(s_1: b_1, s_2: b_3, s_3: b_6, s_4: b_4)$. This match represents that the first detected object is matched to the BIM object with element index is 1, and the second, third, and last detected objects are matched to the BIM objects with element indexes 3, 6 and 4, respectively. Since the objects in S_{object} are located in the odometry frame which varies depending on the robot starting position and there is an unknown transformation between the odometry frame and the BIM frame, various example embodiments found that absolute object coordinate information cannot be used to match the two sets of objects. However, despite the transformation between two frames, various example embodiments found that the distance between two object and the rotation angle from one vector to another are the same in the two odometry and BIM frames. Therefore, according to various example embodiments, the relative positions among the objects are leveraged to find feasible matches between detected objects and BIM objects.

[0093] For any pair of BIM objects b_u and $b_v \in B_{object}$ (as defined in Equation (5)) with x_u, x_v , and y_u, y_v being the x and y coordinates for the two pair of BIM objects respectively, the relative distance (relative distance information) between them in the BIM frame may be represented by d_{uv}^B as:

5

$$d_{uv}^B = \sqrt{(x_v - x_u)^2 + (y_v - y_u)^2}$$

(Equation 6)

[0094] Similarly, for any pair of detected objects s_i and $s_j \in S_{object}$, the relative distance between them in the odometry frame may be denoted as $(v_{ij,x}^S, v_{ij,y}^S)$.

10 [0095] For the same pair of BIM objects b_u and b_v , the relative direction vector (relative direction information) in the BIM frame $(v_{uv,x}^B, v_{uv,y}^B)$ may be defined as:

$$(v_{uv,x}^B, v_{uv,y}^B) = \left(\frac{x_v - x_u}{d_{uv}^B}, \frac{y_v - y_u}{d_{uv}^B} \right)$$

(Equation 7)

[0096] Similarly, for any pair of detected objects s_i and $s_j \in S_{object}$, the relative direction
 15 vector in the odometry frame may be denoted as $(v_{ij,x}^S, v_{ij,y}^S)$. Since the relative positions between objects are used to determine a match therebetween, the relative distances and relative directions for all the pairs of objects may be calculated for both object sets (S_{object} and B_{object}). Given any two detected objects (s_i, s_j) and any two BIM objects (b_u, b_v) , the object labels and the distance between the objects are used as a criteria to compare the sequence of the two
 20 detected objects (which forms a two-object pattern) with the sequence of two BIM objects (which forms a two-object pattern). To match the two detected objects with two BIM objects, various example embodiments first determine whether their object classes are matched (i.e., whether $label(s_i) = label(b_u)$ and $label(s_j) = label(b_v)$) and then determine whether the relative distance between the two detected objects (s_i, s_j) matches with the relative distance
 25 between the two BIM objects (b_u, b_v) .

[0097] In various example embodiments, to accommodate localization errors caused by the fluctuation of sensor measurements, a match is taken as feasible when the relative distance error between the detected objects and the BIM objects is within a predefined threshold. For example, a valid match may be defined with a function z where $z(\{s_i, b_u\}, \{s_j, b_v\}) = 1$ represents a valid
 30 match. The function z for two-object matching may thus be defined as:

$$z(\{s_i, b_u\}, \{s_j, b_v\}) = \begin{cases} 1, & \text{if } |d_{uv}^B - d_{ij}^S| < \alpha \\ 0, & \text{otherwise} \end{cases}$$

(Equation 8)

where α is the threshold for the error in the relative distance.

[0098] In various example embodiments, the above criteria to match a pair of objects is extended to match a sequence of three objects (i.e., three-object matching). In this regard, the sequence of three objects may thus define a three-object pattern. To match a sequence of detected objects (s_i, s_j, s_k) to a sequence of BIM objects (b_u, b_v, b_w) , in various example embodiments, the three-object pattern (e.g., triangular pattern) that is formed by the detected objects is matched to a corresponding object pattern in the BIM. In this regard, in addition to the criteria for label matching and relative distance matching as described above for the two-object matching, direction (or angle) matching is further considered in the three-object matching. For a group of three objects (b_u, b_v, b_w) , a rotation angle (θ_{uvw}) (rotation angle information of the group of BIM objects (b_u, b_v, b_w)) from direction vector v_{uv}^B to v_{uw}^B in the BIM frame may be defined as:

$$(\theta_{uvw}) = \text{atan2}\left(\frac{v_{uv,x}^B v_{uw,y}^B - v_{uv,y}^B v_{uw,x}^B}{v_{uv,x}^B v_{uw,x}^B + v_{uv,y}^B v_{uw,y}^B}\right)$$

(Equation 9)

[0099] Similarly, a rotation angle from v_{ij}^S to v_{ik}^S for detected objects (s_i, s_j, s_k) in the odometry frame may be calculated and represented by θ_{ijk} (rotation angle information of the group of detected objects (s_i, s_j, s_k)). In this regard, various example embodiments advantageously determine and utilize the rotation angles such that the sequence of the detected objects are considered to opt-out false matches with BIM objects. For example, the matched BIM object sequences (b_u, b_v, b_w) and (b_u, b_w, b_v) have the same object pattern geometrically as shown in FIG. 15, but represent different travel paths, leading to one estimation that the robot started at b_u and currently located at b_w , while another estimation that the path ends at b_v . By considering the rotation angles according to various example embodiments of the present invention, given the rotation angle from the first direction vector, v_{ij}^S to the second direction vector, v_{ik}^S is anti-clockwise for the detected objects, the matched BIM object sequence can only be (b_u, b_v, b_w) . Therefore, the criteria for the matching of three objects is defined by combining the criteria for two-object matching and the criteria for the conformance of rotation angles. Accordingly, assuming $z(\{s_i, b_u\}, \{s_j, b_v\}, \{s_k, b_w\}) = 1$ represents a feasible match, z may be defined as:

$$z(\{s_i, b_u\}, \{s_j, b_v\}, \{s_k, b_w\}) = \begin{cases} 1, & \text{if } z(\{s_i, b_u\}, \{s_j, b_v\}) = 1, z(\{s_i, b_u\}, \{s_k, b_w\}) = 1, |\theta_{uvw} - \theta_{ijk}| < \beta \\ 0, & \text{otherwise} \end{cases} \quad (\text{Equation 10})$$

where α is the predetermined threshold for the error in the distance measurement and β is the predetermined threshold for the difference in the measurement of direction (or angle).

5 **[00100]** In various example embodiments, the above technique to match three objects (i.e., the three-object matching) is applied to determine a complete match for a sequence of detected objects, $m : S_{object} \rightarrow B_{object}$, that matches each object $s_i (i = 1, 2, 3, \dots n^S)$ in S_{object} to a corresponding BIM object denoted as $m(s_i)$. In particular, in various example embodiments, matching a sequence of detected objects (S_{object}) comprises matching a sequence of groups of
 10 detected objects (e.g., a group of three detected objects, and thus, the above three-object matching) of the sequence of detected objects with respect to the plurality of BIM objects to obtain, for each group of the sequence of groups of detected objects, one or more matching sub-sequences of BIM objects for the group of detected objects based on a matching condition (e.g., Equation (10)) with respect to the group of detected objects. In this regard, the one or more
 15 matching sub-sequences of BIM objects obtained for the group of detected objects based on the matching condition is based on object classification information and object position-based information of the group of detected objects and object classification information and object position-based information of the one or more matching sub-sequences of BIM objects. By way of an example only and without limitation, an object matching algorithm (Algorithm 1)
 20 according to various example embodiments of the present invention is shown in FIG. 16.

[00101] In various example embodiments, the matching process is performed using a bottom-up approach. Given that the first detected object in S_{object} is s_1 , all the BIM objects in the BIM list that have the same label as s_1 are selected as possible candidates. From the possible matches for the first detected object, it is then determined whether any remaining BIM object(s)
 25 in the BIM can match with the second detected object s_2 using Equation (8) (i.e., the above-mentioned two-object matching). Then, from all the feasible matches for the detected objects s_1 and s_2 , each possible two-object combination is evaluated to add a third BIM object with respect to the sequence of detected objects s_1, s_2 and s_3 that satisfies the match criteria given in Equation (10). For each subsequent detected object in S_{object} , a group of three detected
 30 objects is formed with the immediately two previous objects and the matching criteria for three-object matching (Equation (10)) is applied to obtain one or more matching groups (sub-

sequences) of BIM building objects for the group of three detected objects. This continues iteratively until all the detected objects in S_{object} are matched with BIM objects in B_{object} .

[00102] Accordingly, each subsequent group with respect to an immediately preceding group of the sequence of groups of detected objects partially overlaps with the immediately preceding group of detected objects. In this regard, for the above-mentioned each subsequent group of detected objects, the one or more matching sub-sequences of BIM objects obtained for the subsequent group of detected objects based on the match condition (e.g., Equation (10)) is further based on the one or more matching sub-sequences of BIM objects obtained for the immediately preceding group of detected objects. More particularly, the above-mentioned each subsequent group of detected objects comprises: an immediately subsequent detected building object of the sequence of detected objects with respect to a last detected object of the immediately preceding group of detected objects, and all detected objects of the immediately preceding group of detected objects except a first detected object thereof. Similarly, for each of the one or more matching sub-sequences of BIM objects obtained for the subsequent group of detected objects, the matching sub-sequence of BIM objects obtained comprises: a BIM object matching the immediately subsequent detected object of the subsequent group of detected objects, and all BIM objects of a related matching sub-sequence of the one or more matching sub-sequences of BIM objects obtained for the immediately preceding group of detected objects except a last BIM object thereof.

[00103] Therefore, in various example embodiments, matching the sequence of detected objects comprises determining the one or more matching sequences of BIM objects for the sequence of detected objects, each of the one or more matching sequences of BIM objects being determined based on a corresponding matching sub-sequence of BIM objects obtained for a last group of the sequence of groups of detected objects and based on, for each remaining group of the sequence of groups of detected objects, a related matching sub-sequence of the one or more matching sub-sequences of BIM objects obtained for the group of detected objects related to (or linked to) the corresponding matching sub-sequence of BIM objects obtained for the last group of the sequence of groups of detected objects.

[00104] In various example embodiments, the matches in the last iteration are removed if it cannot find any object in the BIM to match the next detected object in the new iteration. Various example embodiments note that there can be many possible matches with fewer objects present, but many false candidates can be discarded as new detected objects are added for matching in the bottom-up matching process. For example, lift and staircase are typically designed to be

close to each other. As an illustrative example, there may be three pairs of lift and staircase along a long corridor and the relative position between them is similar for all three cases. In this scenario, all three feasible matches are generated if the detected objects in S_{object} are a lift and a staircase although two matches are the false candidates. False candidates can be discarded
 5 when more objects are detected that lead previous matches to a dead end. For example, supposing one object pattern is distinguishable from the other two object patterns by confirming that there is a door near to the lift, the match result can thus become unique if a door is observed at the expected location near the lift and added to S_{object} later as the robot continues to explore the environment.

- 10 **[00105]** Accordingly, given a detected object list S_{object} and BIM object list B_{object} , all possible matches from S_{object} to B_{object} can be determined using the object matching method as described hereinbefore according to various example embodiments of the present invention, and by way of an example only, by using the above-mentioned Algorithm 1. Since similar geometric patterns can occur at different locations and thresholds are set for object matching to
 15 accommodate for localization errors, there can be multiple matches in M that satisfy the matching requirements. In various example embodiments, each possible match may be associated with a hypothesis (or an estimation) of the path and the current robot location. In various example embodiments, the match where the BIM object pattern is more similar to the detected object pattern exhibits a higher probability that the match is correct. Accordingly, the
 20 hypotheses (or estimates) of robot locations are not equally probable, with the probabilities higher at selected locations where the detected object pattern (or detected feature pattern) exhibits a higher similarity with the BIM object pattern (or BIM feature pattern). In this regard, in various example embodiments, BIM-based active exploration may then be introduced to accelerate the initialization process by controlling the robot to navigate based on the most
 25 probable estimation and try to find more distinct feature(s) (building object(s)) in the neighborhood to increase the confidence level of the estimation. This is an active exploration technique to exploit the online detection information and BIM information to validate the hypothesis (or estimation) of the robot position and quickly opt-out false matches, instead of spending more time exploring the environment randomly. To facilitate the BIM-based active
 30 exploration, possible matches may be sorted firstly based on the similarity between the detected object pattern and the BIM object pattern, which is quantified by the matching cost that measures the differences between object patterns (or feature patterns).

[00106] In various example embodiments, the evaluation of the match is performed to determine the more probable travel paths and current robot locations. In various example embodiments, based on the criteria for object matching described in relation to Equations (8) and (10), the cost for the match is defined analogically. The cost for the two-feature (two-object) matching considering the relative distance error may be defined as:

$$L_1(\{s_i, b_u\}, \{s_j, b_v\}) = |d_{uv}^B - d_{ij}^S| \quad (\text{Equation 11})$$

and the cost for the three-feature (three-object) matching taking into account both relative distance and relative direction errors is:

$$L_2(\{s_i, b_u\}, \{s_j, b_v\}, \{s_k, b_w\}) = |d_{uv}^B - d_{ij}^S| + |d_{uw}^B - d_{ik}^S| + |\theta_{uvw} - \theta_{ijk}| \quad (\text{Equation 12})$$

[00107] Therefore, in various example embodiments, the total cost of a match m for the detected object list S_{object} may be defined as:

$$J = L_1(\{s_1, m(s_1)\}, \{s_2, m(s_2)\}) + \sum_{i=3}^{n^S} L_2(\{s_{i-2}, m(s_{i-2})\}, \{s_{i-1}, m(s_{i-1})\}, \{s_i, m(s_i)\}) \quad (\text{Equation 13})$$

[00108] Accordingly, the cost function may comprise two parts, namely, the cost for the matches of the first two objects and the cost for the matches of all the subsequent groups of three consecutive objects. For the first two objects, only the cost on the relative distance difference is counted while for the subsequent sequence of three-object groups, both the relative distance difference and relative direction difference are considered. As an example illustration, FIG. 17 shows example costs of matching detected object groups (s_i, s_j, s_k) and (s_i, s_k, s_o) to corresponding BIM objects, whereby the edges and angles that are utilized to calculate the costs for matching the two detected object groups.

[00109] In various example embodiments, the matching problem can also be considered from the perspective of the graph theory. For example, the topological map of the BIM objects can be considered as a complete graph $G(n^B, n^{B^2})$ where each vertex stands for one BIM object. The detected objects form another graph $H(n^S, n^{S^2})$ where each vertex denotes one detected object. The objective is to find an induced subgraph G' of the graph G such that the difference between G' and H is minimized with an objective function given in Equation (13). The rationale of Equation (13) can also be seen as it accounts for the deviation of each edge and the angle

between two directed edges from each vertex. Therefore, it ensures the detected object pattern is matched to a similar object pattern in the BIM map.

Robot positioning and local navigation

5 **[00110]** The estimation of the position of the robot in the BIM frame (at 1112) will now be described in further details according to various example embodiments of the present invention, which corresponds to the position estimating module 212 as described hereinbefore according to various embodiments. In various example embodiments, the object matching outcomes (one or more matching sequences of BIM objects for the sequence of detected building objects) are
10 utilized to estimate the robot position and guide the robot motion planning. In various example embodiments, the estimated position of the robot position may be validated through navigating to nearby target object(s).

[00111] For each feasible match, as the association between the detected objects and BIM objects is determined, locations of the detected objects in the BIM frame are known. In addition,
15 the matched BIM object sequence reflects the robot's trajectory. In various example embodiments, based on the locations of the last (i.e., latest) three detected objects and detection results for the object that the robot is facing, the instant robot location is estimated. The location is calculated through the principle of trilateration based on the distance measurements to the last three objects, as shown in FIG. 18A. In particular, FIG. 18A illustrates robot localization
20 using trilateration.

[00112] Let the coordinates of the objects be (x_i, y_i) and the distance to the object be d_i , where $i = 1, 2, 3$ for the last three detected objects, and the robot location be (x_g, y_g) , the trilateration problem may be formulated as:

$$\begin{aligned} (x_g - x_1)^2 + (y_g - y_1)^2 &= d_1^2 \\ (x_g - x_2)^2 + (y_g - y_2)^2 &= d_2^2 \\ (x_g - x_3)^2 + (y_g - y_3)^2 &= d_3^2 \end{aligned}$$

(Equation 14)

[00113] In various example embodiments, since the distance measurements contain inevitable errors and a single intersection point cannot be found, the robot location (x_g, y_g) may
30 be estimated using the nonlinear least square optimization method to minimize the sum of squares of a set of equations (between the measured distances to the detected objects and the calculated distances based on the estimated position) in the form as:

$$(x_g, y_g) = \arg \min_{x,y} \sum_{i=1}^3 \|d_i^2 - ((x - x_i)^2 + (y - y_i)^2)\|^2$$

(Equation 15)

For example, Equation (3.15) can be solved by the Trust Region Reflective (trf) algorithm using the Python SciPy library.

- 5 **[00114]** As the robot position in the BIM frame (x_g, y_g) is determined, the robot heading direction (or orientation) can then be estimated based on the relative locations between the robot local frame and the global BIM frame, and the locations of the last detected object in these two frames, as shown in FIG. 18B. Suppose the robot yaw angle is θ_g in the BIM frame and hence, the translation and rotation from the BIM frame to the robot frame are x_g, y_g, θ_g , respectively.

- 10 Let the object location in the robot frame be (o_{rx}, o_{ry}) . The coordinates of the last detected object in the BIM frame (o_{bx}, o_{by}) can be expressed as:

$$\begin{bmatrix} o_{bx} \\ o_{by} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_g & -\sin \theta_g & x_g \\ \sin \theta_g & \cos \theta_g & y_g \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} o_{rx} \\ o_{ry} \\ 1 \end{bmatrix}$$

(Equation 16)

- [00115]** Since the object coordinates in the robot frame and the BIM frame are known and the robot position in the BIM frame (x_g, y_g) is determined, Equation (16) may be rearranged, taking $\cos \theta_g$ and $\sin \theta_g$ as the variables, into:

$$\begin{bmatrix} o_{rx} & -o_{ry} \\ o_{ry} & o_{rx} \end{bmatrix} \begin{bmatrix} \cos \theta_g \\ \sin \theta_g \end{bmatrix} = \begin{bmatrix} o_{bx} - x_g \\ o_{by} - y_g \end{bmatrix}$$

(Equation 17)

- [00116]** Then, $\cos \theta_g$ and $\sin \theta_g$ can be solved with the linear equations, and θ_g can be determined as:

$$\theta_g = \text{atan2}\left(\frac{\sin \theta_g}{\cos \theta_g}\right) = \text{atan2}\left(\frac{(o_{by} - y_g)o_{rx} - (o_{bx} - x_g)o_{ry}}{(o_{bx} - x_g)o_{rx} + (o_{by} - y_g)o_{ry}}\right)$$

(Equation 18)

- [00117]** Therefore, for each feasible match, the corresponding robot pose, (x_g, y_g, θ_g) can be calculated or estimated according to various example embodiments of the present invention.
- 25 As there are more than one feasible match, multiple estimated robot locations may be obtained after performing the object matching. However, depending on the total deviation cost of each match as defined in Equation (13), the probabilities that the robot is located at certain positions are higher than other positions. While the number of feasible matches is less than a threshold

m_{thre} , it indicates that there are very few feasible matches and there is a high possibility that the match with the least cost is the true match. To accelerate the localization process, various example embodiments control the robot to navigate based on the most probable position estimation and attempt to detect nearby target feature(s) (object(s)) to increase the confidence level of the estimation.

[00118] The approach of local navigation utilizes the BIM information to plan local navigation efficiently such that the robot quickly finds other objects and opts out wrong candidates of object matching accordingly instead of wasting time exploring the environment randomly. The nearby objects with distances to the robot smaller than a threshold are sorted by the priority of category first and then sorted by the distance to the robot. A category that contains fewer objects exhibits a higher priority since the objects are more distinct and more false matches can be removed if the object is detected. For example, given two lifts and twenty doors in the testing environment, the category, ‘lift’ is given a higher priority since the number of possible objects the robot is observing becomes two if a lift is detected while it is twenty if a door is detected. Therefore, to find a nearby object that raises the confidence level for the estimation of robot location, the object that is closest to the robot in the category with the highest priority is selected as the target object.

[00119] In various example embodiments, a BIM-based local navigation method (at 1116) is performed to control the robot to approach the target object. For example, the Bug navigation algorithm is a well-known navigation algorithm that is suitable for unknown environments with proven termination conditions. It works in situation where locations and shapes of obstacles in the environment are unknown but the target’s relative position is available. By way of an example only and without limitation, assuming that the location of the target object is (x_t, y_t) and the robot location is (x_g, y_g) , a modified Bug algorithm for the indoor environment with long corridors is provided in Algorithm 2 shown in FIG. 19, where $d_f, d_l, d_r, d_{fl}, d_{fr}$, represent the distances to nearest obstacles in the front, left, right, front left, and front right regions of the robot respectively, and Δ_{yaw} is the error between the desired yaw angle and the robot yaw angle. θ_1 and θ_2 are the tolerances for the yaw error while d_{min} is the threshold for the allowable distance to the front obstacle. These three tolerance parameters may be defined by a user. The desired yaw angle yaw_d and yaw error Δ_{yaw} are calculated where (x, y, yaw) is the robot pose at the instant time:

$$yaw_d = atan2\left(\frac{y_t - y}{x_t - x}\right)$$

(Equation 19)

$$\Delta_{yaw} = yaw_d - yaw$$

(Equation 20)

5 **[00120]** Firstly, the robot rotates to the direction that points to the target object location and then navigates to the target point. When the robot navigates to the target location, it goes straight while the yaw error is within a predefined threshold. It rotates and adjusts the yaw angle if it deviates from the planned straight path connecting the robot pose and the target location. Once an obstacle is observed in the front, the robot avoids the obstacle by switching to the status of
10 wall following. Depending on the orientation of the robot to the facing wall, either the left wall following or the right wall following is performed. The robot follows the wall that reduces the distance between the robot and the target point as it goes along. On the way the robot follows the wall, once reaching the end of a wall and meeting the condition that there is no obstacle between the robot and the target, the robot state is switched from wall following to navigating
15 to the target point. For illustration purposes, an example scenario that illustrates the robot local navigation according to various example embodiments is shown in FIG. 20.

[00121] In various example embodiments, in scenarios where other objects are detected when navigating towards the target object, the robot may stop and record the object information into the detected object list S_{object} . As a new object is added and the feasible matches between
20 detected objects and BIM objects are updated with false candidates discarded, the most probable match may change accordingly. Accordingly, a new/updated target point may be determined based on the latest most probable match and the robot performs local navigation to the new/updated target point.

[00122] Accordingly, in various example embodiments, validating a matching sequence of
25 BIM objects may include: selecting a target BIM object from a plurality of BIM objects; controlling the robot to navigate to a target location in the building with respect to the target BIM object; detecting, using the image sensor and the trained object detector, one or more new objects in the building as the robot navigates to the target location to obtain an updated sequence of detected objects including the one or more new objects; matching the updated sequence of
30 detected objects with respect to the plurality of BIM objects to obtain an updated matching sequence of BIM objects for the updated sequence of detected objects; and updating the estimated position of the robot in the BIM frame based on the updated matching sequence of

BIM objects for the updated sequence of detected objects. Furthermore, in various example embodiments, the above-mentioned matching the updated sequence of detected objects includes matching, for each of the one or more new objects in the updated sequence of detected objects, a group (e.g., a new group) of detected objects with respect to the plurality of BIM objects to obtain one or more matching sub-sequences of BIM objects for the group (e.g., the new group) of detected objects based on the matching condition with respect to the group of detected objects, the group of detected objects comprising the new object and a plurality of immediately preceding detected objects with respect to the new object in the updated sequence of detected objects.

10 **[00123]** Various example embodiments note that since there may be errors in robot odometry measurements and robot position using the principle of trilateration, the estimation of the robot position in the BIM frame may not be very accurate. The robot plans its path to the target point based on its relative position to it. In the case that the estimated robot position deviates from the true position, although the robot may manage to navigate for a specific distance and
15 direction, various example embodiments note that the robot may not be able to see the target object at that point due to an offset to the target point. To compensate for such error in robot positioning, in various example embodiments, a recovery action is performed by rotating the robot to extend its view to a wider angle to increase the probability of detecting the target object even though the robot location may not be the target position for detecting the target object.

20 Finally, if no object is detected even after the execution of the recovery action or the robot cannot reach the target position within a time limit t_{thre} , showing that the estimation of the robot position based on the most probable match may be incorrect, the local navigation terminates. The robot may then switch to the random exploration mode to search for other objects.

[00124] Accordingly, in various example embodiments, local navigation is an active
25 localization process by utilizing online detection information (detected objects) and BIM information (BIM objects) to validate the hypothesis or estimation of the robot position and accelerate the initialization process. The robot may plan its path based on its estimated positions and the confidence of each estimation. Through local navigation, the robot may try to find more distinct objects to quickly opt-out false matches.

30

Online decision making

[00125] Based on various components of the robot initialization system 400 provided according to various example embodiments of the present invention, the robot may continue to

explore the environment, detect objects, match the detected objects and executes local navigation when necessary until the only possible match is found. By way of an example only and without limitation, an overall decision-making process for robot initialization (e.g., coarse initialization) is provided in Algorithm 3 shown in FIG. 21, where $S_{current}$, $S_{windows}$, S_{object} are the sets of detected objects for the current image, the past sequence of images, and the confirmed objects since the initialization process starts, respectively, as explained hereinbefore, M is a list or set that maintains all the possible matches between the detected objects in S_{object} and BIM objects in BIM (B_{object}) and P is a list or set of possible robot poses in the BIM frame corresponding to possible matches in M .

- 10 **[00126]** Algorithm 3 takes BIM information (BIM objects) and sensor data (detected objects) as input and performs decision-making for robot motion and execution of programs in sub-systems. Initially, the robot is deployed at an unknown location in the operating environment and it conducts random exploration to search for objects. While exploring the unknown environment, objects are detected and the filtering of detected objects in the image stream is applied to avoid false or duplicated recordings of detected objects. Once three or more objects have been added to the confirmed object set S_{object} , the association between detected objects and BIM objects can be established and hypotheses (estimation) of the robot pose are feasible.
- 15 **[00127]** Considering that similar geometric relationships of a group of objects may appear at different locations within the working environment, multiple probable matches can be obtained. All the possible matches may be passed to M and the list of corresponding estimated robot poses P may be determined. In the case that only one feasible match exists, the robot location can be determined directly and the initialization process is completed. While multiple matches are provided in M , the match with the highest confidence is adopted and the estimated robot location (x_g, y_g) is taken to validate the hypothesis. The nearest object location (x_t, y_t) is determined using the robot location (x_g, y_g) and BIM information. Afterward, local navigation is performed with a target point (x_t, y_t) . In the process of the local navigation, if a new object is detected, the detected object set S_{object} is updated and the new target point for local navigation is generated according to the latest M and P lists. On the other hand, supposing no object is detected until a pre-defined time limit t_{thre} , the recovery action is carried out to extend the robot view to a wider angle so that missing detections that may be caused by view occlusion can be minimized or avoided. Moreover, if there is no object detected even after the execution of the recovery action, this may indicate that the target point may not be achievable
- 20
- 25
- 30

and the estimation of the robot pose being validated is incorrect. Thereafter, the robot state may be switched back to the random exploration mode to find other objects.

[00128] The robot initialization process is terminated when a unique match is found. Then the confirmed robot position and heading direction can be determined using Equations (15) and (18). For example, this may then complete a first (e.g., coarse) initialization where the robot movement route and the robot position (e.g., preliminary or rough position) are obtained.

Multi-level positioning rectification

[00129] In various example embodiments, since there may be errors in the measurements of robot odometry, noise in depth measurements and deviation between the bounding box centers and object centers, the accuracy of the first (e.g., coarse) robot localization result that relies on the odometry data and object detection results can be improved. In this regard, various example embodiments provide multi-level positioning rectification (at 1118) to improve the robot positioning accuracy and facilitate the autonomous robot navigation. FIG. 22 depicts a schematic block diagram showing the multi-level positioning rectification 1118 according to various example embodiments of the present invention. As shown in FIG. 22, the robot pose estimated by trilateration result may first be subjected to BIM-based rectification 2004 based on BIM information of geometric boundaries/constraints. For example, an estimated position that is inside a closed/inaccessible region according to BIM information may be shifted to an open/accessible region. In various example embodiments, to further improve localization accuracy, particle filter-based fine tuning 2008 may be applied to the above rectified robot pose. In various example embodiments, the multi-level positioning rectification 1118 may take the robot pose rectified based on BIM information as an initial estimation and positioning tracking may then be applied to fine tune the initial estimation until the covariance of robot pose distribution is lower than a predefined threshold.

[00130] In various example embodiments, BIM information for all the walls may be retrieved to provide boundary information of the reachable/accessible areas. Based on the start position, end position and the surface norm of the walls, the perpendicular distance and the nearest point on the wall to the estimated robot pose are obtained. The nearest wall to the robot is then determined by sorting the distances. Since the robot follows the wall with a predefined threshold d_{thre} , any robot pose estimated with a smaller distance to the wall is unrealistic and BIM-based rectification 2004 based on geometric constraints is applied to rectify the robot pose to be sufficiently away from the walls.

[00131] FIGs. 23A and 23B depict schematic drawings illustrating example multi-level positioning rectifications according to various example embodiments of the present invention. In this regard, FIGs. 23A and 23B depict the corridor and boundaries, the last detected object, the estimated robot position and the rectified position. As the distance between the estimated robot position (x_g, y_g) and the wall is d_{wall} , which is smaller than the distance threshold, the estimated location is adjusted to accessible regions first by keeping an allowable distance (e.g., predetermined distance) to the wall. To obtain the rectified robot pose, two adjusted points that are on the perpendicular line to the wall, and yield a distance of d_{thre} to the wall, are first determined as intermediate candidate positions.

[00132] Let the wall surface norm retrieved from BIM to be $\vec{n}_w$, the wall directional vector pointing from (x_w, y_w) to the other end of the wall can be calculated and expressed as $\vec{r}_w$. The coordinates of the nearest point on the wall to the estimated robot position, (x_n, y_n) can be determined using the vector projection equation and line equation. Firstly, the vector from (x_w, y_w) to (x_n, y_n) can be calculated by projecting vector $(x_g - x_w, y_g - y_w)$ along the wall. Then, (x_n, y_n) is determined using the definition of line equation with the start point and the line directional vector as:

$$(x_n, y_n) = (x_w, y_w) + \frac{\vec{r}_w \cdot (x_g - x_w, y_g - y_w)}{\|\vec{r}_w\|} \vec{r}_w \quad (\text{Equation 21})$$

[00133] Then, the coordinates of the two adjusted points can be determined based on the ratio of d_{wall} over d_{thre} as:

$$\begin{aligned} x_{p1} &= x_n + (x_g - x_n) \frac{d_{thre}}{d_{wall}} \\ x_{p2} &= x_n - (x_g - x_n) \frac{d_{thre}}{d_{wall}} \\ y_{p1} &= y_n + (y_g - y_n) \frac{d_{thre}}{d_{wall}} \\ y_{p2} &= y_n - (y_g - y_n) \frac{d_{thre}}{d_{wall}} \end{aligned}$$

[00134] As there are two adjusted candidate points for each piece of wall, the point that is within the corridor is selected while the one beyond the boundaries of the corridor is discarded. Since the wall surface norm $\vec{n}_w$ retrieved from BIM depends on how the wall is modeled, and it can either point towards the corridor or the opposite, and hence $\vec{n}_w$ cannot be used to

determine the point in the corridor directly. Instead, the relative distance and direction between the candidate point and the last detected object 2302 are utilized for such purpose. Let the vectors from the last detected object 2302 to the two points to be $\vec{v}_1$ and $\vec{v}_2$. When the robot is localized close to the wall that is on the other side of the corridor, as shown in FIG. 23A, with the distances to the detected object 2302 $\|\vec{v}_1\| < \|\vec{v}_2\|$, the point that is closer to the detected object 2302, (x_{p1}, y_{p1}) is used to calculate the rectified pose. While the nearest wall and the detected object 2302 are on the same side and the distances to the detected object 2302 are the same for points x_{p1} and (x_{p2}, y_{p2}) , as shown in FIG. 23B, $\vec{v}_1$ and $\vec{v}_2$ are used to calculate the similarity to the norm vector of the object $\vec{n}_o$ and the point with a higher similarity is selected as the candidate point. For the given case in FIG. 23B, as $\vec{n}_o \cdot \vec{v}_1 > \vec{n}_o \cdot \vec{v}_2$, point (x_{p1}, y_{p1}) is selected for the further pose rectification.

[00135] The rectified robot position is then determined based on the candidate point while considering the distance to the last detected object 2302. It is ensured that the distance between the rectified position and the wall is the predefined threshold d_{thre} and the distance to the last detected object 2302 is the measured distance d_o , while the point falls within the corridor. Let the BIM coordinates of the last detected object 2302 be (o_{bx}, o_{by}) , the rectified robot position (x_g^{new}, y_g^{new}) is determined as:

$$\begin{aligned} (x_g^{new}, y_g^{new}) &= \arg \min_{x,y} [(x - x_{p1})^2 + (y - y_{p1})^2] \text{ subject to} \\ \vec{n}_w \cdot (x - x_{p1}, y - y_{p1}) &= 0, (x - o_{bx})^2 + (y - o_{by})^2 = d_o^2 \end{aligned} \quad \text{(Equation 23)}$$

[00136] The corresponding new robot heading θ_g^{new} can be determined with the detection results of the last detected object 2302 using Equation (18). The rectified robot pose using BIM information can be taken as the initial pose for autonomous robot navigation to other target locations directly. In various example embodiments, when there are obstacles or narrow passages, tolerance for the localization error is configured to be smaller in order to avoid collisions. In-place rotation may be performed to recover and adjust the localization results. In various example embodiments, in certain scenarios, to make the navigation smoother and have more accurate localization results, a probabilistic-based method (particle filter-based fine tuning 2008) may be applied to fine tune the pose estimation using the laser sensor observations before passing the initialization result to the navigation stack.

[00137] Accordingly, in various example embodiments, the estimated position of the robot in the BIM frame may be adjusted based on determining that the estimated position of the robot

is within an inaccessible region (inaccessible by the robot) of the building based on geometric constraint information extracted from the BIM of the building. Furthermore, in various example embodiments, the above-mentioned adjusting the estimated position of the robot in the BIM frame comprises: extracting a nearest BIM wall to the robot at the estimated position from the BIM of the building; and adjusting the estimated position of the robot based on a predetermined spacing to the nearest BIM wall and a detected distance of the robot to a latest object detected.

[00138] In various example embodiments, the ROS AMCL package may be used for the fine tuning of the pose estimation by applying robot position tracking. The robot positioning tracking using probabilistic algorithms may be variants of the Bayes filter and the application is called Markov localization which transforms a probabilistic belief at time $t-1$ into the belief at time t . The problem is formulated in the way that the robot pose x_t at time step t is a random variable and $x_t = [x, y, \theta]^T \in \mathbb{R}^2 \times \mathbb{S}$. Taking the map m as input, the belief of the robot pose $bel(x_t)$ is updated recursively incorporating the motion model and observation model as:

$$bel(x_t) = \eta p(z_t|x_t, m) \int p(x_t|u_t - x_{t-1}, m) bel(x_{t-1}) dx$$

(Equation 24)

where η is a normalization constant, z_t and u_t are the observation and control inputs at time t .

[00139] The Monte Carlo Localization (MCL) is an application of Markov localization using a particle filter to track the pose of a robot against a known map. For example, the ROS AMCL package implementing Adaptive Monte Carlo Localization with KLD sampling may be used for the fine tuning of the pose estimation. The robot is controlled to follow the wall and move from the stopping position of coarse positioning for another few meters, given the output from BIM-based rectification 2004 as an input for the AMCL algorithm. The robot pose is updated for each time step with the control inputs and the Lidar readings. As the robot moves, the particle cloud becomes denser and denser, reflecting the pose estimation converges and the confidence of the belief in robot pose increases. The robot stops once a statistics bond on the localization error is obtained and the output from AMCL is the fine-tuned robot pose.

[00140] Accordingly, a BIM-based robot initialization system 400 using object detection is provided according to various example embodiments of the present invention. The integrated robot initialization system 400 utilizes both the offline data and online information to achieve automated and infrastructure-free robot initialization. In the robot initialization system 400, BIM information (BIM objects) is retrieved to provide knowledge of the environment while the CNN-based object detector is trained and used to acquire online visual information (detected building objects). The robot may explore the environment to find as many distinct visual

features (objects) as possible. Object matching is performed to investigate the association between the detected objects and BIM objects. The robot pose is estimated based on the matching results and then it may then be combined with BIM information to facilitate active local navigation, which helps to accelerate the initialization process. Once the only possible match is found, the robot is localized using the principle of trilateration with the location information of the last (latest) three detected objects. After the initial or coarse localization, the multi-level positioning rectification is applied to compensate for the localization error using BIM geometric information and the particle filter based position tracking technique.

10 Experimental Results and Discussions

[00141] To demonstrate the robustness and effectiveness of the robot initialization system 400 as described herein according to various example embodiments, it is implemented and validated at a university building and also residential blocks at ongoing construction sites.

15 *Experimental setup*

[00142] An example mobile platform (Scout robot) was used to develop and test the robot initialization system 400. The mechatronic architecture and hardware of the example robot is shown in FIG. 24. Scout robot is a four-wheel differential drive, skid-steering robot with zero-degree turning radius, driven by 4x200 Watt brushless servo motors. The onboard processing is achieved by Nvidia Jetson AGX Xavier and a mounted industrial PC which is equipped with a NVIDIA GeForce RTX 2080 graphics card. The Ubuntu 18.04 LTS operation system runs ROS onboard the Jetson computer, and the real-time detection is performed using the industrial PC. The robot is equipped with (has mounted thereon) a variety of perception sensors: a 3D lidar (RoboSense RS-LiDAR-16), an RGBD camera (Intel D455 RealSense), IMU, and wheel encoders. The lidar and motion sensors are used to facilitate robot navigation, and the RGBD sensor is used to capture images of scenes in a surrounding environment for subsequent detection and localization of building objects.

[00143] The robot initialization system 400 was first validated in hallways at the university building and then tested in residential buildings under construction. Two construction blocks with different floor plans at the site were selected for the experiments. All testing environments exhibit the characteristics of highly symmetric settings with similar plain walls and pillars. Example scenes of the testing environments, namely, the university building, residential block A and residential block B, are shown in FIGs. 25A, 25B and 25C, respectively.

Experiments at university building

[00144] As mentioned above, the robot initialization system 400 was first tested in a hallway environment at the university building. The whole floor was 189.13 m long and 21.15 m wide.

- 5 There were two main narrow corridors with 2 m wide and a few paths connecting the two long corridors. The robot was deployed at random locations along the hallway and the aim was to find its position autonomously in the environment through exploring the environment and relating the sensory information to the known BIM information.

10 *Offline configuration for the environment*

[00145] A 3D BIM model was created first using Autodesk Revit. The locations and dimensions of building objects (or architectural components) including doors, staircases and lifts were measured and those building objects were built in the BIM based on the measured geometry information. The 3D view of the testing environment, i.e., the BIM model for the hallway, is shown in FIG. 26. The floor region shaded in grey was the feasible area for the robot to travel around and the regions in white surrounded by the walls were offices and were considered as unreachable regions.

- [00146]** In the testing environment, there are 38 doors, 5 lifts and 5 staircases. Each building object (or component/element) is considered a distinctive feature. The distribution of the building objects in the hallway is shown in a floor plan for the hallway in FIG. 27. The whole floor is divided into three sections and the object IDs for each section are given as well. The semantic and geometric information for the total 48 building objects was retrieved automatically with the developed program using Dynamo software. BIM information included the object/element ID, object/element category, coordinates, norm vector of the surface plane, width and height, as presented in Table 2 shown in FIG. 28. Since the doors to staircases were normally closed, for experimental purposes only and without limitation, only doors and lifts were selected as building objects while staircases were excluded for this set of experiments to consider the real situations and validate the robustness of the robot initialization system 400 with fewer types of building objects available.

30

Online robot initialization

[00147] The robot was initially deployed at 17 randomly selected locations along the hallway, as shown in FIG. 29. In particular, FIG. 29 shows various robot starting positions and captured

scenes of the hallway. The dots and arrows represent the robot's starting positions and the heading directions. Pictures of the environment were captured at various locations and they are included in FIG. 29. There are repetitive corridors with plain walls and less distinctive geometric features in the university building, demonstrating its similar characteristics to the construction environments. As the robot started exploring the environment, it followed the decision-making algorithm (the above-mentioned Algorithm 3) described hereinbefore according to various example embodiments. The robot followed the left side wall and stopped to record building object locations if doors or lifts were detected using the trained CNN-based object detector. When three or more building objects were detected, object matching was performed to investigate the categories of detected building objects and geometric relationships between building objects and concluded the valid matches between detected building objects and BIM building objects using the above-mentioned Algorithm 1 described hereinbefore according to various example embodiments.

[00148] As described hereinbefore, a valid match between detected building object sequence and the BIM building object sequence was defined with tolerance to accommodate errors in object localization due to accumulated errors in odometry data and the error in bounding box coordinates of the object detection and depth measurements. The thresholds that reflected such tolerance for the object matching, α and β in Equations (8) and (10) were tuned to ensure that the ground truth match could be found with the least number of false candidate matches added to the list of possible matches. The measured distances and angle between detected building objects (s_i, s_j, s_k) and the ground truth values for corresponding BIM building objects (b_u, b_v, b_w) and measured angle errors were recorded for 94 sets of measured data in a separate series of experiments. Based on the obtained experimental data, it was observed that there was a linear trend between ground truth distances between BIM building objects and the distance errors in object matching, as shown in FIG. 30A. In particular, FIG. 30A shows the relationships between BIM building object distances and the measured distance errors. Moreover, the angle error became larger when the distances between building objects were smaller and the ground truth angles were larger, as shown in FIG. 30B. In particular, FIG. 30B shows the relationships between BIM building object distances and relative angles between BIM building objects. Therefore, α and β were defined considering the ground truth distances (d_{uv}^B) and angles (θ_{uvw}) between BIM building objects instead of arbitrary constants. The tuned thresholds for the object matching are defined as:

$$\alpha = \begin{cases} 2, & \text{if } d_{uv}^B < 10 \\ 2 + (d_{uv}^B - 10) \times 0.04, & \text{if } d_{uv}^B \geq 10 \end{cases}$$

(Equation 25)

$$\beta = \begin{cases} 10, & \text{if } d_{uv}^B < 10, \theta_{uvw} < 90 \\ 10 - (d_{uv}^B - 10) \times 0.1, & \text{if } d_{uv}^B \geq 10, \theta_{uvw} < 90 \\ 15, & \text{if } d_{uv}^B < 10, \theta_{uvw} \geq 90 \\ 15 - (d_{uv}^B - 10) \times 0.1, & \text{if } d_{uv}^B \geq 10, \theta_{uvw} \geq 90 \end{cases}$$

(Equation 26)

- 5 **[00149]** With the tuned matching threshold and the matching algorithm, the robot explored the environment and matched detected building objects to the corresponding building objects until the unique match was determined. Once the match was found, the robot pose was estimated through the principle of trilateration with the locations of the last three building objects. Among the 17 starting locations shown in FIG. 29, the correct matched building object
- 10 sequences and the estimations of robot pose were found for all the tests. The starting positions, the number of detected building objects to localize the robot, moving distances for the coarse localization and the corresponding predicted robot routes to find the positions are presented in Table 3 shown in FIG. 31. In particular, Table 3 shows building object matching results for different robot starting positions at the hallway.
- 15 **[00150]** On average the robot traveled for 44.88 m to find its location. The position was found after 3 or 4 building objects were detected in general and more building objects may be needed if a similar pattern of building objects appeared at multiple locations. It was shown that the number of feasible matches reduced as more building objects were detected since false matching candidates were filtered out. FIG. 32 shows a graph showing the variations in the
- 20 number of feasible matches with the number of building objects detected. The minimum, maximum, and average number of feasible matches are shown on each bar. As can be seen, when 3 building objects were detected, more than 10 possible robot positions were found on average. As 4 and more building objects were detected, the average count of possible locations was reduced to less than 5.
- 25 **[00151]** Taking starting position P3 as an example, FIG. 33 shows the process of filtering out wrong matching candidates for feasible matches for 3, 4, 5, 6 detected building objects, respectively. In particular, FIG. 33 shows possible robot positions and paths with different numbers of building objects detected, namely, 3, 4, 5, 6 building objects, respectively. When 3 building objects were detected, 26 feasible matches were generated. All the possible starting
- 30 points are shown as circle dots in the floor plan. The 8 matches with the least matching cost are

shown with arrows representing the robot path. The square dots represent the last detected building object for the 8 matches. As the robot continued to explore, the fourth building object was detected and the number of possible matches was reduced to less than 10. Five selected travel paths and the ending building objects are shown. The unique matching was determined with 6 building objects detected.

[00152] BIM-based local navigation was implemented to locally tune the robot path such that more distinctive building object can be found to accelerate the initialization process and avoided the large matching error. As discussed in hereinbefore, when there were 5 or fewer feasible matches with the detected building object, the match with the least cost was selected to examine the hypothesis. Nearby distinctive building objects to the estimated robot pose were searched and the robot tried to navigate to the target point where new building objects can be seen. Taking P13 as an example, when the first 3 door objects were detected, there were 5 feasible matches of BIM building objects which were [62938, 105595, 105512], [105512, 105544, 105595], [66749, 66781, 65905], [106230, 66812, 66749], [66781, 66812, 65905] respectively. Since the first match [62938, 105595, 105512] exhibited the highest similarity with the detected building object pattern, the robot position was estimated with the coordinates of those three BIM building objects. Based on the robot position, it was found that there was supposed to be a lift nearby, which was the lift with ID 96688, as shown in FIG. 34. Instead of continuing to follow the wall, the robot navigated to the lift using the Algorithm 2 as described hereinbefore according to various example embodiments. As the robot approached the target point, the lift with ID 96688 was detected and added to the detected building object list or set. With the new building object detected, a single feasible match was found and the other 4 matching candidates were filtered out. Such a BIM-based local navigation approach utilized the BIM information to advantageously control the robot to find more distinctive building objects quickly to narrow down the possible matches and reduce the exploration time.

[00153] It was observed that if similar object patterns appeared at multiple locations and there were no distinctive nearby building objects, the robot traveled for a long distance to determine the unique match. For position P17 where the robot started at one end of the corridor, the robot followed the left side wall and moved along the long corridor. Building objects that were more distinctive such as lifts were not seen but only doors were observed. Since the distances between every two consecutive doors along the corridors were similar in this building, making the object patterns to be similar along the corridor, it was difficult to find a unique

match. After the robot moved for around 100 meters and turned into the lobby area and recognized the lifts, a unique match was finally found.

[00154] For the 17 tests, the robot ending pose can be determined using equations introduced in section “Robot positioning and local navigation” described hereinbefore based on the BIM coordinates of the last three detected building objects and the relative position between the last
5 coordinates of the last three detected building objects and the relative position between the last (latest) detected building object and the robot. BIM-based constraint analysis was applied to rectify the coarsely estimated robot pose and ensure that the robot was not inside or extremely close to any wall. For the tests where rectified localization results were relatively accurate, as shown in FIGs. 35A to 35B, the estimated positions could be passed explicitly to the robot
10 navigation stack guiding the robot to navigate to the target point smoothly and further adjusting the timely localization while navigating. In particular, FIGs. 35A to 35C show the BIM-based rectification result at the hallway for the ground truth, the coarse localization and the BIM-based rectification, respectively.

[00155] In various example embodiments, PF-based fine tuning may be applied to further
15 improve the localization accuracy for the results with a larger error in pose estimation, as shown in FIGs. 36A to 36D. In particular, FIGs. 36A to 36D show the PF-based fine tuning result at the hallway for the ground truth, the coarse localization, the BIM-based rectification and the PF-based fine tuning, respectively.

[00156] The moving distance for coarse localization, the total distance with the fine tuning
20 applied, and localization errors for all the 17 tests are summarized in Table 4 shown in FIG. 37. The error in robot coordinates and yaw angle between the ground truth and the estimated pose for the coarse localization and multi-level rectification are presented. It was shown that before the BIM-based rectification, the average error in robot positioning and robot yaw angle was 1.32 m and 22.63 degrees respectively. The average errors were reduced to 0.61 m and 8.91
25 degrees after the rectification with the applied geometric constraints, and improved to 0.21 m and 1.47 degrees with the PF-based fine tuning that required additional robot movement. Considering the width and length of the robot base were 0.7 m and 0.95 m, the errors in robot positioning after the BIM-based rectification and fine tuning were around 50% and 20% of the diagonal of the robot base, respectively.

30

Experimental evaluations

[00157] The robot initialization method (which simply be referred to herein as the present initialization method) according to various example embodiments was compared with two

commonly used robot global localization methods to demonstrate the robustness and efficiency of the present initialization method. The marker-based method using ArUco tag and laser-based method, AMCL global localization were implemented to localize the robot from the same 17 starting positions using the same wall following based motion planning algorithm. Various
 5 criteria were applied to evaluate the initialization performance, including the success rate, average moving distance, and average localization error that reflected the robustness, efficiency, and accuracy respectively. The recorded moving distance was the total distance the robot traveled to localize itself. The same criteria for successful localization was used for both the present initialization method, and the two comparative methods. The localization was
 10 considered as failed if the moving distance was larger than 150 m but the position could not be found yet, or the localization error was larger than 5 m.

[00158] To implement the marker-based method, two ArUco tags were deployed for each section and the location was the exact middle of each corridor segment. Therefore, there were six markers in total, evenly distributed throughout the entire working environment. The 20 cm
 15 x 20 cm markers were printed on paper and pasted on the walls, as shown in FIG. 38. FIG. 38 shows the marker-based localization including marker deployment and marker detection. The RGBD camera was used to detect the marker using OpenCV library. The black boundary was detected to localize the marker and the inner binary pattern was decoded to determine the corresponding unique marker ID. As the four corners of the marker were detected, the
 20 transformation between the camera frame and the maker frame that took the center of the marker as origin can be determined by solving the Perspective-n-Point (PnP) problem. Given the marker location in the BIM frame, the robot position and heading direction were calculated accordingly. FIG. 38 also shows the marker detection results, including the marker ID, marker location and yaw angle in the camera frame, and the drawn marker frame. FIG. 39 shows Table
 25 5 presenting the localization results for all the 17 points, indicating whether the localization was successful ('Y' stands for succeeded and 'N' stands for failed) and the number of markers passed before the location was found. The moving distance and localization error are also presented in Table 5.

[00159] Besides the marker-based method, AMCL global localization mode was tested to
 30 localize the robot without any prior knowledge about the position. The sample localization process is depicted in FIGs 40A to 40B. In particular, FIGs. 40A to 40B show the robot localization process with AMCL for randomly distributed particles, multiple particle clouds, confirmed particle cloud and final converged particle cloud, respectively. Initially, 5000

particles were randomly generated and uniformly distributed on the map. As the robot moved, the probabilistic distribution of the robot pose was updated with the motion model and observation model. The particles were grouped into multiple particle clouds and each cloud represented one possible located region. At this stage, the covariance of the robot pose was large, indicating it was in the process of localizing. When the robot moved further, ambiguities can be eliminated gradually, and then, there was only one particle cloud left, showing that the rough location was acquired. The localization confidence level increased as the moving distance increased and the localization was considered completed when the covariance was lower than the predefined threshold, which was the same as that for PF-based fine tuning in the present initialization method. The robot localization results with AMCL are presented in Table 6 shown in FIG. 41.

[00160] The success rate, average moving distance and localization errors for the present initialization method with BIM-based rectification, both BIM-based rectification and PF-based fine tuning, as well as the two baseline methods (marker-based (ArUco tag) localization method and laser-based (AMCL) localization method) are presented in Table 7 shown in FIG. 42A. Only the successful trials were used to calculate the average moving distance and localization error. The initialization results for all the 17 starting positions using the present BIM-based initialization method with different levels of rectification, and the two comparable methods are presented in FIGs. 42B to 42D. In particular, FIGs. 42B to 42D show plots of experimental evaluations with comparative baseline methods for: moving distance (FIG. 42B), translational localization error (FIG. 42C), and rotational localization error (FIG. 42D).

[00161] It can be seen that the present initialization method with BIM-based rectification yielded the highest success rate of 100% while the ArUco marker method and AMCL method achieved the rate of 94% and 35% respectively. There was localization failure for one point with the ArUco tag method as the robot detected the marker from a far distance of over 10 m, but the marker size was only 20 cm and thus occupied just a few pixels in the image. There were large errors in the detection of corner locations and the prediction of the robot pose. In a study of pose estimation accuracy using markers, it was concluded that there was a clear error dependence on the distance to the markers and positioning accuracy was severely degraded as observation distance was larger than 4 m. Regarding the AMCL method, the success rate was low since the particle filter based method performed well for the environments exhibiting sufficient unique features but poorly for repetitive environments containing similar structures. For the experiments at the university building hallway, it failed for most of the starting positions,

especially for the ones starting in the long corridor where no distinctive geometric features can be observed. For such cases, the correct particle cloud was always lost while tracking the robot position. As similar geometric features (building objects) appeared at various locations, although the particles often converged to a single pose hypothesis quickly, AMCL was vulnerable to wrong localization to other positions exhibiting similar characteristics, as shown in FIGs. 43A and 43B. In particular, FIGs. 43A and 43B show localization problems with AMCL method including wrong localization (FIG. 43A) and inaccurate localization (FIG. 43B).

[00162] In terms of efficiency, localization using AMCL was the fastest with the shortest moving distance as it implemented a probabilistic based algorithm that retrieved information from each time step and updated the position estimation continuously. It was capable of converging quickly. Since the present initialization method and marker-based method both required the detection of discrete visual objects, the moving distance was larger and the moving distance also depended on how far every two features were separated. It was also observed that the moving distance with PF-based fine tuning applied was about 5 m longer than localization with just BIM-based rectification since robot movement was required for the fine tuning process. Despite the level of positioning rectification applied, the moving distances were slightly lower than the marker-based method. It was noticed that the robot missed the first marker for two trials due to the poor viewpoints and change of illumination as the robot moved. Therefore, it took a long distance to find the second marker if the first one encountered was not detected.

The present initialization method that detected common architectural features (building objects) outperformed since it collected the information of all detected objects and utilized the information of relative positions. Although some objects along the way were not detected, it could still rely on the other detected objects to localize, rather than moving for a long time to find the unique marker. Moreover, both the marker-based and laser-based methods were passive methods where robot motion was not configured for the localization task. The present initialization utilized the BIM geometric and semantic information to guide the robot motion and performed local navigation and thus, managed to expedite the localization process and eliminate the wrong position candidates efficiently.

[00163] Regarding the localization error, the marker-based method was most accurate in position estimation, with an error of 17 cm, while AMCL method had the lowest heading error of 1.43 degrees. It showed that PF-based fine tuning greatly reduced the localization error compared with the method merely applied with BIM-based rectification. The present initialization method with fine tuning had comparable accuracy with the best performed

methods in terms of both positioning and heading, giving 21 cm and 1.47 degrees error respectively. It was also noticed that the variation in positioning accuracy was large for AMCL, with some trials possessing a sub-cm error while others having 2 to 4 meters error. The reason was that although the robot could correctly find the region it fell in, there could be large offsets between the ground truth and estimated location in the corridor areas lacking sufficient building objects to rectify and align the estimation, as shown in FIG. 43B. The boundness of error was not guaranteed for global localization and it was influenced by the initial particle distribution. In contrast, the present initialization method with PF-based fine tuning applied the same AMCL algorithm for the rectification of localization but obtained more accurate positioning results since the coarse estimation of the pose was given by visual feature (building object) matching, rather than a random distribution.

[00164] Accordingly, the present initialization method using BIM based rectification and PF-based fine tuning was the most robust in the sense of achieving a higher success rate and was applicable for all the starting points while the other two baseline methods failed for certain scenarios. The present initialization method and marker-based method have a relatively larger moving distance than AMCL global localization method. The present initialization method with PF-based fine tuning possessed an average localization error of 0.21 m and 1.47 degrees, which were comparable with the most accurate positioning and heading results obtained from baseline methods. Poor success rate and a large average positioning error had been seen with AMCL global localization since wrong localization and large offsets presented in repetitive environments. Marker-based method provided acceptable localization accuracy but required additional effort on marker deployment and was prone to error with viewpoint and environment changes. Compared with the baseline methods, the present initialization method was infrastructure-free and robust, providing accurate localization results. However, it was found that the moving distance for certain scenarios with similar feature patterns was relatively long. Nevertheless, the present initialization method was able to handle scenarios with many ambiguities due to similar feature patterns, thereby demonstrating the robustness and accuracy of the present initialization method.

Implementations at construction environments

[00165] The present initialization method was implemented at the real active construction sites to test its viability to automate the robot positioning and navigation for construction automation. Two residential blocks undergoing construction were selected for the experiments.

[00166] The two blocks had the same appearance of architectural objects, which were lifts, staircases and main gates of the units. There were the same numbers of building objects for the two blocks, with 13 main gates, 3 lifts and 2 staircases on one floor for each block. However, the layouts for the two blocks were different, making the geometric patterns among the above-mentioned building objects different. The layouts and scenes of the two environments are shown in FIGs. 44A and 44B and FIGs. 45A and 45B, and the lengths of the corridors were 61.3 and 62.8 m respectively. In particular, FIG. 44A shows a BIM model of Block A and FIG. 44B shows robot starting positions at Block A. Similarly, FIG. 45A shows a BIM model of Block B and FIG. 45B shows robot starting positions at Block B. Both testing environments were repetitive with long corridors, symmetric plain walls and pillars. Block A was in the final inspection stage while Block B involved more construction exercises, leaving more construction materials and moving workers in the corridor. The real industrial BIM models were provided by the project's main contractor and the models were used to retrieve the geometric and semantic information of the building objects.

[00167] In the online initialization tests, the robot was placed at 9 evenly distributed points along the corridors for each block, as shown in FIGs. 44B and 45B. The parameters tuned for object matching using the university building testing results in Tables 2 and 3 were adopted and it performed well and generated the correct object matching and route prediction for all the tests. The testing robot localization results for Block A and Block B are presented in Table 8 and Table 9 shown in FIGs. 46 and 47, respectively, and the results of multi-level localization are depicted in FIGs. 48A to 48D. In particular, FIGs. 48A to 48D show the robot multi-level localization result for the ground truth, the coarse localization, the BIM-based rectification and the PF-based fine tuning, respectively.

[00168] The robot started from the deployed position, explored the environment and matched detected objects to the corresponding BIM objects using the present online decision-making strategy until the unique match was determined. Then the pose of the robot was estimated and rectified using Equations (15), (18), and Equations (23), (24). Among the 35 starting locations in the three environments, the correct matched object sequences and the estimations of robot pose were found for all the experiments. The number of detected objects and moving distance for the coarse initialization, localization errors, the total moving distance after applying the PF-based fine tuning and the corresponding localization errors for different environments are presented in FIGs. 49A to 49F and the mean values are summarized in Table 10 shown in FIG. 50. In particular, FIGs. 49A to 49F show plots of experimental results for

different environments: number of detected objects for coarse initialization (FIG. 49A), moving distances for coarse and fine initialization (FIG. 49B), translational localization error in the university building environment (FIG. 49C), translational localization error in residential construction environments (FIG. 49D), rotational localization error in the university building environment (FIG. 49E) and rotational localization error in residential construction environments (FIG. 49F).

[00169] The robot localization statistics of the experimental results for different environments in terms of mean values are presented in Table 10 shown in FIG. 50. Similar to the university building tests, the robot location and route can be predicted after detecting 3 to 7 visual features (building objects) for building construction site environments. However, the moving distance for coarse localization and fine localization were around 15 m and 22 m respectively, which were less than half of those for university building environments as the distance between main gates at the residential blocks was around 5 m while it was 10 to 20 m between doors at university building corridors. Moreover, the positioning errors for the residential blocks were lower in general, especially for the localization with BIM-based rectification as the corridors and lobby area at the residential blocks were more narrow, providing more geometric constraints. Comparing the residential blocks, a larger number of visual features (building objects) and moving distances were seen to localize the robot at Block B due to two reasons. Firstly, Block B had a more symmetric layout around the lift lobby, making it difficult to confirm in which direction the robot approached the lifts for starting positions p1, p2, p3 and p4. Unlike Block B, Block A had distinct geometric patterns between building objects around the lobby area and therefore, localization was quick for starting positions p6, p7, p8 and p9. Secondly, Block B had more dynamic obstacles in the corridor, blocking the robot's view or forcing the robot to change the path to avoid obstacles and thus, causing missed detection of objects along the way and longer moving distances.

[00170] The feasibility of the robot initialization system was also validated by performing robot navigation that took the localization result as the initial pose. Navigation tests were conducted for the 9 trails at Block B and selected points at the university building environment. It was shown that the localization results with BIM-based rectification were sufficient to start navigating to the target location. However, for the cases where heading errors were around 10 degrees or above, the robot could be stuck during the navigation as the accumulated pose estimation error caused failure in path planning. The navigation recovery was required to rectify the robot pose by in-place rotation and obstacle clearance on the map, and may cause absorption

of the goal finally. Such a problem could be addressed by implementing the PF-based fine tuning in the positioning stage according to various example embodiments and then performing the navigation.

5 [00171] The testing results at residential blocks validated the generality of the robot initialization method at different environments. The developed system and tuned parameters in the university building environment could be utilized directly in the residential blocks. It also showed that the present initialization method was a general solution to be applied across the residential blocks with different layouts at the same construction site since the object detector and robot motion planning algorithm were environment-independent. It provided the ease of
10 robot initialization without any marker and infrastructure deployment, or the construction of the unique image or signal strength database for robot initialization in each different construction environment (e.g., each individual block).

[00172] For example, after the coarse initialization, the BIM-based constraint analysis and PF-based fine tuning were applied to rectify the coarsely estimated robot pose. The robot
15 localization result using this multi-level positioning rectification was illustrated in FIGs. 36A to 36D. As presented in FIGs. 49C to 49F, the multi-level positioning rectification significantly improved the robot localization accuracy. In the university building environment, the average translational and rotational errors for the coarse initialization were 1.32 m and 22.63 degrees, respectively. The average errors were reduced to 0.61 m and 8.91 degrees after the rectification
20 with the applied BIM geometric constraints and the accuracy was improved to 0.21 m and 1.47 degrees with the PF-based fine tuning. In residential construction environments, the mean localization errors were reduced from 0.61 m and 13.35 degrees to around 0.3 m and 5 degrees with BIM-based rectification and 0.15 m and 2 degrees with PF-based fine tuning. Compared with the university building environment, the positioning errors for residential blocks were
25 lower in general, especially for the localization with BIM-based rectification as the corridors and lobby area at residential blocks were more narrow, providing more geometric constraints. The feasibility of the robot initialization system was further validated by performing robot navigation that took the fine tuned initialization result as the initial pose. Navigation tests were conducted for the 9 trails at Block B and selected points at the university building environment.
30 The robot was able to navigate to the target point smoothly with the given initialization. The testing results at different environments validated the generality of the robot initialization method in highly symmetric construction environments, including both industrial buildings like the university building and residential buildings. Moreover, the BIM information retrieval

application, the object detector and the robot motion planning algorithm were found to be environment-independent. The robot initialization system was applied for the two residential blocks of different layouts and construction work without any additional pre-scanning. The experiments illustrated that the robot initialization system can be easily applied across different blocks and is useful for the robot platforms that are designed to work at multiple blocks at the same construction site. It provided the ease of robot initialization without any marker and infrastructure deployment, or the construction of the unique image or signal strength database.

[00173] Accordingly, the experimental results of the BIM-based indoor robot initialization method according to various example embodiments using object detection at different environments, including a university building and two residential blocks under construction are shown. The robot was able to find its position autonomously and initialize itself in the map for 17 trials at university building corridors and 9 trails for each residential block. The present initialization method provided an infrastructure-free robot initialization solution that utilized BIM information and object detection, and released the effort of device deployment and placement configuration required by traditional robot positioning methods. The method can be employed for different types of construction environments and was applicable for various blocks at the same site with the reusable object detector and BIM information processor. Compared with baseline marker-based and laser-based localization methods, the present initialization method achieved a higher success rate and possessed comparable localization accuracy with the most accurate results obtained by the other methods in terms of the average error in positioning and robot heading. The laser-based global localization method allowed faster initialization, but suffered from the high failure rate.

[00174] While embodiments of the invention have been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A method of mobile robot initialization in a building based on a building information model (BIM) of the building, the method comprising:
- 5 obtaining, from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building;
- detecting a plurality of building objects of the building from the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects;
- 10 matching the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and
- estimating a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of
- 15 detected building objects.
2. The method according to claim 1, wherein
- said matching the sequence of detected building objects comprises matching a sequence of groups of detected building objects of the sequence of detected building objects with respect
- 20 to the plurality of BIM building objects to obtain, for each group of the sequence of groups of detected building objects, one or more matching sub-sequences of BIM building objects for the group of detected building objects based on a matching condition with respect to the group of detected building objects, and
- the one or more matching sub-sequences of BIM building objects obtained for the group
- 25 of detected building objects based on the matching condition is based on object classification information and object position-based information of the group of detected building objects and object classification information and object position-based information of the one or more matching sub-sequences of BIM building objects.
- 30 3. The method according to claim 2, wherein
- said object classification information of the group of detected building objects comprises, for each detected building object of the group of detected building objects, object classification information of the detected building object,

said object position-based information of the group of detected building objects comprises, for each pair of a plurality of pairs of detected building objects of the group of detected building objects, relative distance information and relative direction information of the pair of detected building objects,

5 said object classification information of the one or more matching sub-sequences of BIM building objects comprises, for each BIM building object of each of the one or more matching sub-sequences of BIM building objects, object classification information of the BIM building object, and

10 said object position-based information of the one or more matching sub-sequences of BIM building objects comprises, for each pair of a plurality of pairs of BIM building objects of each of the one or more matching sub-sequences of BIM building objects, relative distance information and relative direction information of the pair of BIM building objects.

4. The method according to claim 3, wherein

15 said object position-based information of the group of detected building objects further comprises rotation angle information of the group of detected building objects determined based on the relative direction information of each pair of the plurality of pairs of detected building objects of the group of detected building objects, and

20 said object position-based information of the one or more matching sub-sequences of BIM building objects further comprises, for each of the one or more matching sub-sequences of BIM building objects, rotation angle information of the matching sub-sequence of BIM building objects determined based on the relative direction information of each pair of the plurality of pairs of BIM building objects of the matching sub-sequence of BIM building objects.

25

5. The method according to claim 3 or 4, wherein

each subsequent group with respect to an immediately preceding group of the sequence of groups of detected building objects partially overlaps with the immediately preceding group of detected building objects; and

30 for said each subsequent group of detected building objects, the one or more matching sub-sequences of BIM building objects obtained for the subsequent group of detected building objects based on the match condition is further based on the one or more matching sub-

sequences of BIM building objects obtained for the immediately preceding group of detected building objects.

6. The method according to claim 5, wherein

5 said each subsequent group of detected building objects comprises: an immediately subsequent detected building object of the sequence of detected building objects with respect to a last detected building object of the immediately preceding group of detected building objects, and all detected building objects of the immediately preceding group of detected building objects except a first detected building object thereof, and

10 for each of the one or more matching sub-sequences of BIM building objects obtained for the subsequent group of detected building objects, the matching sub-sequence of BIM building objects obtained comprises: a BIM building object matching the immediately subsequent detected building object of the subsequent group of detected building objects, and all BIM building objects of a related matching sub-sequence of the one or more matching sub-
15 sequences of BIM building objects obtained for the immediately preceding group of detected building objects except a last BIM building object thereof.

7. The method according to claim 5 or 6, wherein said matching the sequence of detected building objects further comprises determining the one or more matching sequences of BIM
20 building objects for the sequence of detected building objects, each of the one or more matching sequences of BIM building objects being determined based on a corresponding matching sub-sequence of BIM building objects obtained for a last group of the sequence of groups of detected building objects and based on, for each remaining group of the sequence of groups of detected building objects, a related matching sub-sequence of the one or more matching sub-sequences
25 of BIM building objects obtained for the remaining group of detected building objects related to the corresponding matching sub-sequence of BIM building objects obtained for the last group of detected building objects.

8. The method according to any one of claims 2 to 7, wherein each group of the sequence
30 of groups of detected building objects has three detected building objects.

9. The method according to any one of claims 2 to 8, wherein

the one or more matching sequences of BIM building objects for the sequence of detected building objects comprises a plurality of matching sequences of BIM building objects for the sequence of detected building objects, and

5 said estimating the position of the mobile robot in the BIM frame is based on a matching sequence of the plurality of matching sequences of BIM building objects that has a least matching error amongst a plurality of matching errors determined for the plurality of matching sequences of BIM building objects, respectively.

10 10. The method according to any one of claims 2 to 9, further comprises validating the matching sequence of BIM building objects comprising:

 selecting a target BIM building object from the plurality of BIM building objects;

 controlling the mobile robot to navigate to a target location in the building with respect to the target BIM building object;

15 detecting, using the image sensor and the trained building object detector, one or more new building objects in the building as the mobile robot navigates to the target location to obtain an updated sequence of detected building objects including the one or more new building objects;

 matching the updated sequence of detected building objects with respect to the plurality of BIM building objects to obtain an updated matching sequence of BIM building objects for
20 the updated sequence of detected building objects; and

 updating the estimated position of the mobile robot in the BIM frame based on the updated matching sequence of BIM building objects for the updated sequence of detected building objects.

25 11. The method according to claim 10, wherein

 said matching the updated sequence of detected building objects comprises matching, for each of the one or more new building objects in the updated sequence of detected building objects, a group of detected building objects with respect to the plurality of BIM building objects to obtain one or more matching sub-sequences of BIM building objects for the group of
30 detected building objects based on the matching condition with respect to the group of detected building objects, the group of detected building objects comprising the new building object and a plurality of immediately preceding detected building objects with respect to the new building object in the updated sequence of detected building objects.

12. The method according to claim 10 or 11, wherein the target BIM building object is selected from the plurality of BIM building objects based on a distance of the target BIM building object to the position of the mobile robot and a classification of the target BIM building object.

13. The method according to any one of claims 2 to 12, further comprising:
adjusting the estimated position of the mobile robot in the BIM frame based on determining that the estimated position of the mobile robot is within an inaccessible region of the building based on geometric constraint information extracted from the BIM of the building.

14. The method according to claim 13, wherein said adjusting the estimated position of the mobile robot in the BIM frame comprises:
extracting a nearest BIM wall to the mobile robot at the estimated position from the BIM of the building; and
adjusting the estimated position of the mobile robot based on a predetermined spacing to the nearest BIM wall and a detected distance of the mobile robot to a latest building object detected.

15. A system for mobile robot initialization in a building based on a building information model (BIM) of the building, the system comprising:
at least one memory; and
at least one processor communicatively coupled to the at least one memory and configured to:

obtain, from an image sensor mounted on a mobile robot, a plurality of images of scenes in the building as the mobile robot navigates in the building;

detect a plurality of building objects of the building from the plurality of images obtained using a trained building object detector to obtain a sequence of detected building objects;

match the sequence of detected building objects with respect to a plurality of BIM building objects extracted from the BIM of the building to obtain one or more matching sequences of BIM building objects for the sequence of detected building objects; and

estimate a position of the mobile robot in a BIM frame of the BIM based on a matching sequence of the one or more matching sequences of BIM building objects for the sequence of detected building objects.

5 16. The system according to claim 15, wherein

said match the sequence of detected building objects comprises matching a sequence of groups of detected building objects of the sequence of detected building objects with respect to the plurality of BIM building objects to obtain, for each group of the sequence of groups of detected building objects, one or more matching sub-sequences of BIM building objects for the
10 group of detected building objects based on a matching condition with respect to the group of detected building objects, and

the one or more matching sub-sequences of BIM building objects obtained for the group of detected building objects based on the matching condition is based on object classification information and object position-based information of the group of detected building objects and
15 object classification information and object position-based information of the one or more matching sub-sequences of BIM building objects.

17. The system according to claim 16, wherein

said object classification information of the group of detected building objects
20 comprises, for each detected building object of the group of detected building objects, object classification information of the detected building object,

said object position-based information of the group of detected building objects comprises, for each pair of a plurality of pairs of detected building objects of the group of detected building objects, relative distance information and relative direction information of the
25 pair of detected building objects,

said object classification information of the one or more matching sub-sequences of BIM building objects comprises, for each BIM building object of each of the one or more matching sub-sequences of BIM building objects, object classification information of the BIM building object, and

30 said object position-based information of the one or more matching sub-sequences of BIM building objects comprises, for each pair of a plurality of pairs of BIM building objects of each of the one or more matching sub-sequences of BIM building objects, relative distance information and relative direction information of the pair of BIM building objects.

18. The system according to claim 17, wherein

said object position-based information of the group of detected building objects further comprises rotation angle information of the group of detected building objects determined based on the relative direction information of each pair of the plurality of pairs of detected building objects of the group of detected building objects, and

said object position-based information of the one or more matching sub-sequences of BIM building objects further comprises, for each of the one or more matching sub-sequences of BIM building objects, rotation angle information of the matching sub-sequence of BIM building objects determined based on the relative direction information of each pair of the plurality of pairs of BIM building objects of the matching sub-sequence of BIM building objects.

19. The system according to claim 17 or 18, wherein

each subsequent group with respect to an immediately preceding group of the sequence of groups of detected building objects partially overlaps with the immediately preceding group of detected building objects; and

for said each subsequent group of detected building objects, the one or more matching sub-sequences of BIM building objects obtained for the subsequent group of detected building objects based on the match condition is further based on the one or more matching sub-sequences of BIM building objects obtained for the immediately preceding group of detected building objects.

20. The system according to claim 19, wherein

said each subsequent group of detected building objects comprises: an immediately subsequent detected building object of the sequence of detected building objects with respect to a last detected building object of the immediately preceding group of detected building objects, and all detected building objects of the immediately preceding group of detected building objects except a first detected building object thereof, and

for each of the one or more matching sub-sequences of BIM building objects obtained for the subsequent group of detected building objects, the matching sub-sequence of BIM building objects obtained comprises: a BIM building object matching the immediately subsequent detected building object of the subsequent group of detected building objects, and

all BIM building objects of a related matching sub-sequence of the one or more matching sub-sequences of BIM building objects obtained for the immediately preceding group of detected building objects except a last BIM building object thereof.

5 21. The system according to claim 19 or 20, wherein said match the sequence of detected building objects further comprises determining the one or more matching sequences of BIM building objects for the sequence of detected building objects, each of the one or more matching sequences of BIM building objects being determined based on a corresponding matching sub-
10 sequence of BIM building objects obtained for a last group of the sequence of groups of detected building objects and based on, for each remaining group of the sequence of groups of detected building objects, a related matching sub-sequence of the one or more matching sub-sequences of BIM building objects obtained for the remaining group of detected building objects related to the corresponding matching sub-sequence of BIM building objects obtained for the last group of detected building objects.

15

22. The system according to any one of claims 16 to 21, wherein each group of the sequence of groups of detected building objects has three detected building objects.

23. The system according to any one of claims 16 to 22, wherein
20 the one or more matching sequences of BIM building objects for the sequence of detected building objects comprises a plurality of matching sequences of BIM building objects for the sequence of detected building objects, and
 said estimate the position of the mobile robot in the BIM frame is based on a matching sequence of the plurality of matching sequences of BIM building objects that has a least
25 matching error amongst a plurality of matching errors determined for the plurality of matching sequences of BIM building objects, respectively.

24. The system according to any one of claims 16 to 23, wherein the at least one processor is further configured to validate the matching sequence of BIM building objects comprising:
30 selecting a target BIM building object from the plurality of BIM building objects;
 controlling the mobile robot to navigate to a target location in the building with respect to the target BIM building object;

detecting, using the image sensor and the trained building object detector, one or more new building objects in the building as the mobile robot navigates to the target location to obtain an updated sequence of detected building objects including the one or more new building objects;

5 matching the updated sequence of detected building objects with respect to the plurality of BIM building objects to obtain an updated matching sequence of BIM building objects for the updated sequence of detected building objects; and

10 updating the estimated position of the mobile robot in the BIM frame based on the updated matching sequence of BIM building objects for the updated sequence of detected building objects.

25. The system according to claim 24, wherein
 said matching the updated sequence of detected building objects comprises matching, for each of the one or more new building objects in the updated sequence of detected building
15 objects, a group of detected building objects with respect to the plurality of BIM building objects to obtain one or more matching sub-sequences of BIM building objects for the group of detected building objects based on the matching condition with respect to the group of detected building objects, the group of detected building objects comprising the new building object and a plurality of immediately preceding detected building objects with respect to the new building
20 object in the updated sequence of detected building objects.

26. The system according to claim 24 or 25, wherein the target BIM building object is selected from the plurality of BIM building objects based on a distance of the target BIM building object to the position of the mobile robot and a classification of the target BIM building
25 object.

27. The system according to any one of claims 16 to 26, wherein the at least one processor is further configured to adjust the estimated position of the mobile robot in the BIM frame based on determining that the estimated position of the mobile robot is within an inaccessible region
30 of the building based on geometric constraint information extracted from the BIM of the building.

28. The system according to claim 27, wherein said adjust the estimated position of the mobile robot in the BIM frame comprises:

extracting a nearest BIM wall to the mobile robot at the estimated position from the BIM of the building; and

5 adjusting the estimated position of the mobile robot based on a predetermined spacing to the nearest BIM wall and a detected distance of the mobile robot to a latest building object detected.

29. A computer program product, embodied in one or more non-transitory computer-
10 readable storage mediums, comprising instructions executable by at least one processor to perform the method of mobile robot initialization in a building based on a BIM of the building according to any one of claims 1 to 14.

30. A mobile robot comprising:

15 a mobile platform configured to navigate in a building;

a sensor system mounted on the mobile platform, the sensor system comprising an image sensor configured to capture a plurality of images of scenes in the building as the mobile robot navigates in the building; and

20 the system according to any one of claims 15 to 28 communicatively coupled to the mobile platform and the sensor system for performing mobile robot initialization in the building based on a BIM of the building.