

**TITLE OF INVENTION: A METHOD AND A SYSTEM FOR UPDATING A STATIC
MAP OF AN ENVIRONMENT**

TECHNICAL FIELD

[0001] This invention relates to a method and a system for updating a map. More particularly, this invention relates to a method and a system for updating a static map of an environment based on sensor data collected by a robot during its operation.

BACKGROUND

[0002] The use of mobile service robots has seen tremendous growth in the past decade. Cleaning robots for example are now extensively being deployed to reduce the physical burden of cleaners in their day-to-day labour-intensive jobs. To perform the job efficiently, the robot usually needs to rely on a stored map for localization and navigation in the environment. However, an operating environment of these robots is subject to change depending on the nature of the environment. If the environment has changed, the stored map may provide inaccurate or wrong guidance to the robot. Moreover, frequent changes of the environment often make it challenging for standard localization algorithms, for example, particle filters to provide an accurate estimation of the robot's global position in the environment. Consequently, there is a need for detecting changes in the environment and updating of the map.

SUMMARY

[0003] According to an aspect of the present disclosure, there is provided a method for updating a static map of an environment. The static map includes a plurality of cells. Each cell of the static map has a respective occupancy status. The method includes obtaining a submap of an area in the environment. The submap is smaller in size than the static map. The submap includes a plurality of cells. Each cell of the submap has a respective occupancy status. The method further includes determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed; and updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap based either on a number of or a percentage of cells of the static map whose occupancy status has changed.

[0004] In some embodiments of the method, determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed includes determining cells of the static map having an occupancy status of unoccupied when a corresponding cell of the submap is identified as a cell having an appearing object and/or determining cells of the static map having an occupancy status of occupied when a corresponding cell of the submap is identified as a cell having a disappearing object.

[0005] In some embodiments of the method, each cell of the static map has an associated signed distance field (SDF). And the corresponding cell of the submap is identified as a cell having an appearing object if the SDF of the corresponding unoccupied cell of the static map is higher than a first threshold.

[0006] In some embodiments of the method, each cell of the submap has an associated signed distance field (SDF) and the corresponding cell of the submap is identified as a cell having a disappearing object if its SDF is higher than a second threshold.

[0007] In some embodiments of the method, obtaining a submap of an area in the environment includes obtaining the submap of the area in the environment by a robot deployed to move along a path in the environment during an operation.

[0008] In some embodiments of the method, updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is carried out if either the number of cells or the percentage of cells exceeds a change threshold.

[0009] In some embodiments of the method, updating occupancy status of the cells of the static map is carried out at the end of the robot's operation, and the generated submap is stored in a memory of the robot only if either the number of cells or the percentage of cells exceeds the change threshold.

[0010] In some embodiments of the method, obtaining the submap of the area in the environment by the robot deployed to move along the path in the environment during the operation includes obtaining multiple scan groups over a time interval as the robot move along the path; processing the multiple scan groups to identify any dynamic object in the area; and generating the submap that excludes any identified dynamic object.

[0011] In some embodiments of the method, the percentage of cells is a first percentage. And the method further includes obtaining a later version of the submap

of the area in the environment; determining cells of the static map corresponding to the cells of the later version of the submap whose occupancy status has changed; obtaining a second percentage of cells of the static map whose occupancy status has changed; and applying Bayesian filtering to the first percentage and the second percentage to obtain an updated percentage. And updating occupancy status of the cells of the static map with occupancy status of the cells of the submap includes updating occupancy status of the cells of the static map with occupancy status of the cells of the submap if the updated percentage exceeds a change threshold.

[0012] In some embodiments of the method, updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is performed autonomously.

[0013] According to another aspect of the present disclosure, there is provided a system for updating a static map of an environment. The static map includes a plurality of cells. Each cell of the static map has a respective occupancy status. The system includes a device that has a processor. The processor is configured to obtain a submap of an area in the environment, wherein the submap is smaller in size than the static map and includes a plurality of cells, each cell of the submap having a respective occupancy status. The processor is further configured to determine cells of the static map corresponding to the cells of the submap whose occupancy status has changed; and update occupancy status of the cells of the static map with the occupancy status of the cells of the submap based either on a number of or a percentage of cells of the static map whose occupancy status has changed.

[0014] In some embodiments of the system, determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed includes determining cells of the static map having an occupancy status of unoccupied when a corresponding cell of the submap is identified as a cell having an appearing object; and/or determining cells of the static map having an occupancy status of occupied when a corresponding cell of the submap is identified as a cell having a disappearing object.

[0015] In some embodiments of the system, wherein each cell of the static map has an associated signed distance field (SDF). And the corresponding cell of the submap is identified as a cell having an appearing object if the SDF of the corresponding unoccupied cell of the static map is higher than a first threshold.

[0016] In some embodiments of the system, each cell of the submap has an associated signed distance field (SDF), and the corresponding cell of the first submap is identified as a cell having a disappearing object if its SDF is higher than a second threshold.

[0017] In some embodiments of the system, obtaining a submap of an area in the environment includes obtaining the submap of the area in the environment by a robot deployed to move along a path in the environment during an operation.

[0018] In some embodiments of the system, updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is carried out if either the number of cells or the percentage of cells exceeds a change threshold.

[0019] In some embodiments of the system, obtaining the submap of the area in the environment based on sensor data provided by the robot deployed to move along the path in the environment during the operation includes obtaining multiple scan groups over a time interval as the robot move along the path; processing the multiple scan groups to identify any dynamic object in the area; and generating the submap that excludes any identified dynamic object.

[0020] In some embodiments of the system, the processor is further configured to obtain a later version of the submap of the area in the environment; determine cells of the static map corresponding to the cells of the later version of the submap whose occupancy status has changed; obtain a second percentage of cells of the static map whose occupancy status has changed; and apply Bayesian filtering to the first and the second percentage to obtain an updated percentage. And updating occupancy status of the cells of the static map with occupancy status of the cells of the submap includes updating occupancy status of the cells of the static map with occupancy status of the cells of the submap if the updated percentage exceeds a change threshold.

[0021] In some embodiments of the system, updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is performed autonomously.

[0022] Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

BRIEF DESCRIPTION OF DRAWINGS

[0023] The invention will be better understood with reference to the drawings, in which:

[Figure 1A] is a flowchart showing a sequence of steps for updating a static map according to an embodiment of the invention;

[Figure 1B] is a flowchart showing a sequence of steps for updating a static map according to another embodiment of the invention;

[Figure 2A] shows an example of the static map in Figure 1A with occupancy information;

[Figure 2B] illustrates how the static map in Figure 2A in the form of an occupancy grid map may be generated;

[Figure 2C] shows a Signed Distance Field (SDF) of the static map in Figure 2A;

[Figure 3] shows a process of creating a set of submaps along a robot's planned path;

[Figure 4A] shows an example of the static map in Figure 1A of an environment a robot is to be operated in;

[Figure 4B] is a map of the environment in Figure 4A captured by the robot as it moves along a planned path, the map showing changes of the environment and a submap generated by the robot at the start of the planned path;

[Figure 4C] shows a Signed Distance Field (SDF) of the static map in Figure 4A;

[Figure 5A] is the static map in Figure 4A showing a number of map portions thereof;

[Figure 5B] is the map in Figure 4B illustrating a process of generating multiple submaps along the planned path;

[Figure 5C] is the Signed Distance Field (SDF) in Figure 4C showing multiple corresponding cells thereof;

[Figure 5D] shows a Signed Distance Field (SDF) of submap 520M in Figure 5B;

[Figure 5E] shows a static map portion 520S in Figure 5A;

[Figure 5F] is the map in Figure 5B showing a Probabilistic Change Detection Map (PCDM) associated with the multiple submaps in Figure 5B;

[Figure 6A] shows a flowchart of a sequence of steps for identifying appearing object(s) in a submap;

[Figure 6B] shows a flowchart of a sequence of steps for identifying a disappearing object(s) in the submap;

[Figure 6C] shows another submap 520M and the corresponding static map portion 520S in Figure 5B; and

[Figure 7] shows a block diagram illustrating an example of a configuration of the robot.

DESCRIPTION OF THE EMBODIMENTS

[0024] Throughout this document, unless otherwise indicated to the contrary, the terms “comprising”, “consisting of”, “having” and the like, are to be construed as non-exhaustive, or in other words, as meaning “including, but not limited to.” Furthermore, throughout the specification, unless the context requires otherwise, the word “include” or variations such as “includes” or “including” will be understood to imply the inclusion of a stated integer or group of integers but not the exclusion of any other integer or group of integers.

[0025] Throughout the description, it is to be appreciated that the term ‘processor/controller’ and its plural form include microcontrollers, microprocessors, programmable integrated circuit chips such as application specific integrated circuit chip (ASIC), computer servers, electronic devices, and/or combination thereof capable of processing one or more input electronic signals to produce one or more output electronic signals. The controller includes one or more input modules and one or more output modules for processing electronic signals.

[0026] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as is commonly understood by a skilled person to which the subject matter herein belongs.

[0027] As shown in the drawings for the purpose of illustration, the invention may be embodied in a method for updating a static map that is less likely to disrupt the operation of a robot when used by the robot for navigation. The static map includes multiple cells. Each of the cells has a respective occupancy status. Generally, the method includes obtaining a submap of an area in the environment. The submap is smaller in size than the static map. The submap also includes multiple cells. Each

cell of the submap also has a respective occupancy status. The method further includes determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed; and updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap based either on a number of or a percentage of cells of the static map whose occupancy status has changed.

[0028] More specifically, Figure 1A is a flowchart showing a sequence 2 of steps in the method for updating a static map of an environment according to an embodiment of the invention. The sequence starts in a “BUILD A STATIC MAP” step S100, wherein a static map of an environment is built. The static map may be built using any means. For example, a robot may be guided to move in the environment to generate a static map of the environment. This static map will be used by the robot to guide its movement during operation, such as but not limited to, a cleaning operation. This static map is stored in a memory of the robot. The static map includes multiple cells. Each cell of the static map includes an occupancy status of the cell. The static map therefore represents the locations of objects in the environment.

[0029] Figure 2A is an example of a static map showing such occupancy status. The static map may be implemented as an occupancy grid map made up of equal-sized squares, hereinafter referred to simply as cells. Each cell holds occupancy status indicating whether an area in the environment corresponding thereto is occupied. For example, the cells 220 in black in the center of the static map in Figure 2A indicate that the corresponding areas in the environment are occupied. In other words, there are objects present in those areas of the environment. The surrounding cells 210 in white indicate that the areas in the environment corresponding thereto are unoccupied. The cell represents a 5cm x 5cm square area in the environment, although it may represent areas of other sizes.

[0030] Figure 2B shows how a robot captures occupancy information to generate the static map in the step S100. The robot is equipped with a sensor, such as a 360-degree distance sensor. When the robot moves in the environment, it uses the sensor to perform a laser range scan. If an object is hit by the laser range scan, an occupancy status of the cell of the static map corresponding to the area where the object is located is set as occupied. If the laser range scan passes through an area, i.e., the laser range scan is not blocked by anything in that area, an occupancy status of the

cells of the static map corresponding to that area is set as unoccupied. For example, when a laser range scan as indicated by a dashed line 230 in Figure 2B hits an object located at an area 240, the cell corresponding to the area 240 is marked as an occupied cell. All the other areas 250 between the robot's location and the occupied area 240 are known to be unoccupied since nothing there blocks the passage of the laser range scan. The occupancy status of the cells corresponding to these unoccupied areas 250 are set as unoccupied.

[0031] Figure 2C shows a Signed Distance Field (SDF) of the static map in Figure 2A. A SDF is a grid-based representation of the geometry of a mapped world where each of its cells stores the signed distance to its nearest occupied cell. Each cell in the SDF represents a cell in the static map. Preferably, the value associated with a cell of the SDF is computed by converting the static occupancy grid map into a binary-valued grid with an occupancy probability threshold differentiating true and false cells, and then taking the Euclidean distance transform of the binary grid with the algorithm described in the article [Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(19):415–428, 2012. doi: 10.4086/toc.2012.v008a019. URL <https://theoryofcomputing.org/articles/v008a019>]. In other words, the value of an unoccupied cell in the SDF represents a distance between the unoccupied cell and a nearest occupied cell. As shown in the static map in Figure 2A, an occupied cell 220 is the nearest occupied cell from an unoccupied cell 210, the distance between the unoccupied cell 220 and the occupied cell 210 is calculated to be $+\sqrt{8}$. Thus, the value of the cell 210V in the SDF is $+\sqrt{8}$ as shown in Figure 2C. Furthermore, occupied cells have an SDF value of -1. The static map is stored in the robot for use during its operation.

[0032] The sequence 2 next proceeds to an "OBTAIN SUBMAP AND DETERMINE OCCUPANCY CHANGES" step S200. Figure 3 illustrates the process of obtaining a set of submaps as the robot moves in the environment. As the robot performs an operation, for example, cleaning the floor, it follows a planned path. While performing the operation along the planned path, the robot will collect data of the environment using its sensor via the laser range scan as described above. As the robot moves along the planned path, the robot collects frames of sensor data at a rate of, for example, ten frames per second. Each frame of sensor data is referred to hereinafter as a scan group. The robot processes these scan groups to identify any dynamic

object in its path. A dynamic object is any object that only temporarily appears in the environment and is distinguished from a stationary object. The robot generates a single submap based on the scan groups. Any identified dynamic object is excluded from the submap. In this manner, as the robot moves along the planned path, a number of submaps can be generated. For example, Figure 3 shows two submaps, S^1 and S^2 , that are created corresponding to robot positions x_t and x_{t+1} along the planned path. Each submap has its own local coordinate system $\{S^1\}$ and $\{S^2\}$ that can be mapped to the overall static map coordinate system $\{M\}$. As described above, each submap S^1 , S^2 is generated based on multiple scan groups to remove dynamic objects in the robot's path. The submaps may or may not overlap one another.

[0033] Figure 4A shows a stored static map of an environment where the robot is to operate along a planned path in. Figure 4B shows a map of the same environment with changes therein. As the robot moves along the planned path, it generates submaps as described above. Figure 4B shows a first submap 410M that is generated corresponding to a cell 410 in the map. For illustration purposes, all the cells in black in Figures 4A and 4B are occupied cells corresponding to areas in the environment occupied by objects. Cells in white are unoccupied cells.

[0034] In Figure 4A, the static map that was first generated shows objects present in the center of the environment forming a T-shape block. Figure 4C shows an SDF of the static map in Figure 4A. As described above, this SDF includes cells having values that indicate whether corresponding cells of the static map are occupied or unoccupied, and if a cell is unoccupied, its value indicates how far that static map cell is from a nearest occupied static map cell.

[0035] When the robot is deployed to operate in the environment, there may be changes in the environment as shown in the map in Figure 4B. As the robot moves along the planned path, it generates submaps as described above for identifying changes in occupancy status in the environment.

[0036] Figure 4B shows the planned path, indicated by a dashed line, for the robot during an operation in the environment. The planned path starts from the cell 410 and ends at a cell 440. A first submap 410M corresponding to the area 410 where the robot was is generated as described above. The submap 410M is a 3x3 square box with nine cells. However, the submap may be of any size. Each cell has an occupancy status that indicates if a corresponding area in the environment is occupied

or unoccupied. Occupied cells are shown in black and unoccupied cells are shown in white in Figures 4A and 4B. In Figure 4B, all nine cells of the submap 410M are white indicating that the area in the environment covered by the submap 410M is unoccupied.

[0037] As the robot moves along the planned path, further submaps are created at predetermined time or distance intervals. Figure 5A is identical to Figure 4A for ease of reference. Figure 5B shows further submaps 520M, 530M generated along the planned path in the actual environment while Figure 5C is identical to Figure 4C again for ease of reference. As shown in Figure 5B, the robot moves from an area corresponding to the cell 410 to another area corresponding to a cell 520 and then to yet another area corresponding to a cell 530 along the planned path. The submaps 410M, 520M and 530M associated with these cells 410, 520 and 530 are generated respectively as described above. Figure 5B merely shows three submaps 410M, 520M, 530M that are generated. It is to be appreciated that other submaps are generated but are not shown in the figure.

[0038] Once each submap 410M, 520M, 530M is generated, the robot processes the submap to identify conflicting cells and consistent cells therein. Conflicting cells may arise due to two possible scenarios. One scenario is when new objects are being added to the environment after the static map was generated. Another scenario is when objects are removed from the environment after the static map was generated. In other words, a conflicting cell is either a cell that is recorded as unoccupied in the static map and is now occupied in the submap, or a cell that is recorded as occupied in the static map and is now unoccupied in the submap.

[0039] Appearing objects refer to objects that were non-existent when the static map was created for the environment but now appear in the environment when the robot is deployed. Figure 6A is a flowchart illustrating the process of identifying appearing objects. Firstly, each cell in a submap 410M, 520M or 530M is checked to identify if it is an occupied cell. If there is at least one occupied cell among the nine cells included in the submap, the appearing objects identification process proceeds to a step A100, otherwise the process ends in a step B100. There are no occupied cells in either the first submap 410M or the second submap 520M, thus the process stops at step B100 when processing these two submaps 410M, 520M. As for the third submap 530M, there are three occupied cells 530-M1, 530-M2 and 530-M3 along a

left column of the submap 530M as shown in Figure 5B and thus the appearing objects identification process proceeds to the step A100.

[0040] In this step A100, the SDF value of a cell in the static map corresponding to each occupied cell in the submap 530M is obtained. The three cells 530-V1, 530-V2, 530-V3 in the static map correspond to the three occupied cells 530-M1, 530-M2, 530-M3 in the submap 530M respectively. The SDF values of these cells 530-V1, 530-V2, 530-V3 are +1, +2 and +2 respectively.

[0041] Next, each of the SDF values is compared with a first threshold in a step A200. If the obtained SDF value is greater than the first threshold, it is determined that no object was present in the original environment when the static map was generated. Hence, the now occupied cell in the submap indicates that an object which was not there previously has now appeared in the current environment. As mentioned above, the SDF value of an unoccupied cell indicates how far the unoccupied cell is from the nearest occupied cell. A small SDF value, such as +1, indicates that the unoccupied static map cell is right next to an occupied cell. A larger SDF value, such as +2, indicates that the unoccupied static map cell is two cells away from the nearest occupied cell. If an object is detected to be in a cell having a larger SDF value, the higher the likelihood that it is a correct detection. If the SDF value of an unoccupied cell of the static map is small, detection of the corresponding submap cell as one with an appearing object may be incorrect because it may be the same object that was already there in a neighbouring occupied static map cell that is detected once again. The first threshold value may for example be 0. In such a case, since all three SDF values of static map cells corresponding to the submap cells 530-M1, 530-M2, 530-M3 are +1, +2 and +2 respectively, they are all above the first threshold of 0. Accordingly, step A310 in Figure 6A is performed, i.e. the occupied cells 530-M1, 530-M2 and 530-M3 in the submap 530M are determined to be cells with appearing object(s). If the obtained SDF value of a static map cell corresponding to an occupied cell in the submap is lower than the first threshold of 0 indicating that the static map cell is occupied, it can be concluded in a step A320 that the occupied submap cell is a consistent cell with no change in its occupancy status. Subsequently, the number of submap cells with appearing objects is counted and recorded for subsequent processing in a step A410. For the submap 530M, there are three cells with appearing objects therein.

[0042] Disappearing objects refer to objects that were present in the environment at the time the static map was created but are no longer present in the current environment. Figure 6B is a flowchart illustrating the process of identifying submap cells with disappearing objects. Firstly, each cell in the static map portions corresponding to those in the submaps 410M, 520M, 530M is checked to identify if there is any occupied cell. If there is any occupied cell in the static map, the disappearing object identification process proceeds to a step C100, otherwise the process ends in a step D100. For the three submaps 410M, 520M and 530M shown in Figure 5B, only the submap 520M has a corresponding cell 520-S1 in the static map that is occupied. The disappearing object identification process is thus further described only with reference to the submap 520M.

[0043] If there is one or more occupied cells in the static map corresponding to a submap, the process continues in a step C100 to obtain the corresponding SDF values of submap cells corresponding to each of the occupied static map cell.

[0044] Each of the obtained SDF values of the submap is compared with a second threshold in a step C200. The second threshold may for example be 0. If the obtained SDF value is above the second threshold, the submap cell is determined to be now unoccupied when it was previously occupied as indicated in the static map. Hence, the unoccupied cell in the submap is identified as a cell with a disappearing object in step C310. For the submap 520M, all cells are unoccupied and thus all the SDF values of the cells of the submap 520M are infinity. Since the submap cell 520-V1 corresponding to the occupied static map cell 520-S1 in Figure 5E has an SDF value of infinity, which is above the second threshold of 0, this cell is identified as one with a disappearing object in the step C310. The cells with disappearing objects are counted in a step C400.

[0045] If the submap cell 520-V1 is occupied (not shown) and has an SDF value of -1 which is lower than the second threshold of 0, then this submap cell is identified as a consistent cell, i.e. a cell having no change in occupancy status in a step C320. Consistent cells are counted and recorded for subsequent processing in a step C420. In the submap 520M shown in Figure 5B, all cells are unoccupied and the SDF value of each cell is infinity. If for example a central cell of that submap 520 is occupied, the SDF values of the submap may be that shown in Figure 6C instead. In such a case, the submap cell 520-V1 has a value of $+\sqrt{2}$ which is still higher than the second

threshold of 0. The submap cell 520-V1 would still be identified as a cell with a disappearing object despite its SDF value being lower now. If, however, the second threshold is raised to a value of 2, then even though the submap cell 520-V1 is unoccupied, it cannot be definitively determined to be so since its neighbouring cell is occupied. In such a case, the submap cell 520-V1 is determined to be a consistent cell instead.

[0046] Consistent cells refer to the submap cells with occupancy status that remains unchanged from the time the static map was created to the time when the submap is generated. These submap cells can either be occupied or unoccupied. Referring to Figure 6A, if no occupancy status change is detected for an occupied submap cell, the occupied submap cell is determined to be a consistent cell. Similarly, if no occupancy status change is detected for an unoccupied submap cell, the unoccupied submap cell is also determined to be a consistent cell.

[0047] With the number of cells with appearing objects, the number of cells with disappearing objects and the number of consistent cells in a submap, a map change percentage for the submap can be calculated based on the following formula:

$$\text{Map Change Percentage} = \frac{\text{Num_Obj_Appear} + \text{Num_Obj_Disappear}}{\text{Num_Obj_Appear} + \text{Num_Obj_Disappear} + \text{Num_Cons_Cells}}$$

wherein

Num_Obj_Appear is the number of cells in the submap with appearing objects;
Num_Obj_Disappear is the number of cells in the submap with disappearing objects; and
Num_Cons_Cells is the number of cells in the submap with unchanged occupancy status.

[0048] The map change percentages obtained for the submaps 410M, 520M and 530M are shown in Table 1 below.

Table 1: Map change percentages

Submap	Num_ Obj_Appear	Num_ Obj_Disappear	Num_Cons_Cells	map change percentage
410M	0	0	9	0

520M	0	1	8	11%
530M	3	0	6	33%

[0049] The sequence in Figure 1A next proceeds to an UPDATE STATIC MAP step S300. In this step S300, the static map is updated based on the map change percentages of the submaps. For example, the static map portion is updated with the corresponding submap if the map change percentage of the submap exceeds a change threshold. This change threshold may be set higher to avoid updating the static map when there is only a small amount of change. For example, change in occupancy status is considered small when furniture is shifted by only a centimeter or when curtains are adjusted to form different fold patterns. Sensor data that is affected by noise in the electronics may also result in only a small amount of change. The change threshold may be set according to the environment.

[0050] The robot 700 may update the static map during its operation or at the end of its operation. If the robot 700 updates the static map during its operation, the robot may update the static map each time a submap is obtained, or it may update the static map after a predetermined number of submaps are obtained. If the robot updates the static map at the end of its operation, submaps whose map change percentages are lower than the change threshold are discarded and not stored; only those submaps whose map change percentages are higher than the change threshold are stored in the robot's memory. This ongoing pruning process to discard submaps reduces the memory requirements of the robot.

[0051] As described above, the steps S100-S300 may be performed by a robot 700 having components shown in Figure 7. The robot 700 includes a processor 710 configured to perform the steps S100-S300, at least one memory module 720, a distance sensor 730, a display 740, an I/O interface 750 and a mobile mechanism 760. The mobile mechanism 760 can be any mechanism which allows the robot 700 to move.

[0052] Advantageously, the above-described method allows map changes to be detected and the map updated autonomously without any human intervention so that the robot's static map is up to date.

[0053] Although the present invention is described as implemented in the above-described embodiment, it is not to be construed to be limited as such. For example,

although the method is described to be using a map change percentage for determining if a map update is to be carried out, the number of cells whose occupancy status has changed may also be used instead.

[0054] As another example, each submap having a first map change percentage is described to be created only once during the robot's operation. However, it is possible that during the same operation the robot may create two or more versions of the same submap. For example, further on in its planned path, the robot may return to the same area where a submap was previously captured. The robot may obtain another version of the submap of the area and determine a second map change percentage for this latest version of the submap. The robot can then apply Bayesian filtering to the first map change percentage and the second map change percentage to obtain an updated map change percentage for the submap, and determine whether to update the corresponding portion of the static map based on this updated map change percentage. This updated map change percentage is likely to be a more accurate reflection of how much the static map portion has really changed. In conjunction with obtaining an updated map change percentage for the submap using Bayesian filtering, the occupancy status of each cell of the submap may also be updated based on applying Bayesian filtering to the corresponding cells in the different versions of the submap.

[0055] As yet a further example, although the method is described to be performed in a robot 700, the method may also be performed by another device, such as but not limited to a robot controller, a mobile phone, and a server, that is data communicatively coupled to the robot.

[0056] In such an embodiment, the robot captures and sends sensor data to the device. The device then performs the method based on the sensor data received from the robot.

[0057] As another example, the steps S100-S200 should not be construed to be limited for use with the step S300. These steps S100-S200 may be used in another method of updating a static map that displays information relating to the map change percentages on the static map to allow a user to select which areas of the static map to update. Figure 1B is a flowchart showing a sequence 4 of steps for performing this method. The steps S100-S200 are the same as those described above and will not be described again. After the map change percentages of the submaps are obtained

in the step S200, the sequence 4 proceeds to an INDICATE CHANGES ON STATIC MAP step S400. In this step S400, the static map is displayed with cells 510, 520, 530 of the submaps 410M, 520M, 530M highlighted accordingly to show the respective map change percentages. For example, the submap 530M has a map change percentage of 33%, the central cell 530 of this submap 530M may be highlighted in red as shown in Figure 5F. The submap 520M has a map change percentage of 11% and its central cell 520 may be highlighted in purple. And the submap 410M has a map change percentage of 0% and its central cell 410 may be highlighted in blue. That is, blue indicates there is no change in occupancy status, red indicates that is a substantial amount of change and shades of purple indicate change amounts between the two. In this manner, the static map is displayed with a heatmap superimposed thereon. Based on such a visual representation of changes in the environment, a user can easily identify which areas have changed after the static map was captured and by how much. The user can then select areas of the static map for updating. The user may select the areas for updating via any input means, including but not limited to, a touchscreen. In this manner, the user has the flexibility to choose certain areas for updating.

[0058] After the user has made his selection, the sequence 4 proceeds to an UPDATE STATIC MAP BASED ON USER'S SELECTION step S500. In this step, only areas in the static map that are selected for updating by the user are updated with the occupancy status in the submaps corresponding to those areas. Once again, the display of the static map may be on the robot's display or a display of a device data communicatively coupled to the robot.

[0059] It should be further appreciated by the person skilled in the art that one or more of the above modifications or improvements, not being mutually exclusive, may be further combined to form yet further embodiments of the present invention.

CLAIMS

1. A method for updating a static map of an environment, wherein the static map includes a plurality of cells, each cell of the static map having a respective occupancy status, the method comprising:

obtaining a submap of an area in the environment, wherein the submap is smaller in size than the static map and includes a plurality of cells, each cell of the submap having a respective occupancy status;

determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed; and

updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap based on one of:

a number of cells of the static map whose occupancy status has changed; and

a percentage of cells of the static map whose occupancy status has changed.

2. The method according to Claim 1, wherein determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed comprises at least one of:

determining cells of the static map having an occupancy status of unoccupied when a corresponding cell of the submap is identified as a cell having an appearing object; and

determining cells of the static map having an occupancy status of occupied when a corresponding cell of the submap is identified as a cell having a disappearing object.

3. The method according to Claim 2, wherein each cell of the static map has an associated signed distance field (SDF), and wherein the corresponding cell of the submap is identified as a cell having an appearing object if the SDF of the corresponding unoccupied cell of the static map is higher than a first threshold.

4. The method according to Claim 2, wherein each cell of the submap has an associated signed distance field (SDF), and wherein the corresponding cell of the

submap is identified as a cell having a disappearing object if its SDF is higher than a second threshold.

5. The method according to Claim 1, wherein obtaining a submap of an area in the environment comprises:

obtaining the submap of the area in the environment by a robot deployed to move along a path in the environment during an operation.

6. The method according to Claim 5, wherein updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is carried out if one of the number of cells and the percentage of cells exceeds a change threshold.

7. The method according to Claim 6, wherein updating occupancy status of the cells of the static map is carried out at the end of the robot's operation, and the generated submap is stored in a memory of the robot only if one of the number of cells and the percentage of cells exceeds the change threshold.

8. The method according to Claim 5, wherein obtaining the submap of the area in the environment by the robot deployed to move along the path in the environment during the operation comprises:

obtaining multiple scan groups over a time interval as the robot move along the path;
processing the multiple scan groups to identify any dynamic object in the area; and
generating the submap that excludes any identified dynamic object.

9. The method according to Claim 1, wherein the percentage of cells is a first percentage and the method further comprises:

obtaining a later version of the submap of the area in the environment;
determining cells of the static map corresponding to the cells of the later version of the submap whose occupancy status has changed;
obtaining a second percentage of cells of the static map whose occupancy status has changed; and

applying Bayesian filtering to the first and the second percentage to obtain an updated percentage;

wherein updating occupancy status of the cells of the static map with occupancy status of the cells of the submap comprises updating occupancy status of the cells of the static map with occupancy status of the cells of the submap if the updated percentage exceeds a change threshold.

10. The method according to Claim 1, wherein updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is performed autonomously.

11. A system for updating a static map of an environment, wherein the static map includes a plurality of cells, each cell of the static map having a respective occupancy status, the system comprising:

a device having a processor configured to:

obtain a submap of an area in the environment, wherein the submap is smaller in size than the static map and includes a plurality of cells, each cell of the submap having a respective occupancy status;

determine cells of the static map corresponding to the cells of the submap whose occupancy status has changed; and

update occupancy status of the cells of the static map with the occupancy status of the cells of the submap based on one of:

a number of cells of the static map whose occupancy status has changed; and

a percentage of cells of the static map whose occupancy status has changed.

12. The system according to Claim 11, wherein determining cells of the static map corresponding to the cells of the submap whose occupancy status has changed comprises at least one of:

determining cells of the static map having an occupancy status of unoccupied when a corresponding cell of the submap is identified as a cell having an appearing object; and

determining cells of the static map having an occupancy status of occupied when a corresponding cell of the is identified as a cell having a disappearing object.

13. The system according to Claim 12, wherein each cell of the static map has an associated signed distance field (SDF), and wherein the corresponding cell of the submap is identified as a cell having an appearing object if the SDF of the corresponding unoccupied cell of the static map is higher than a first threshold.

14. The system according to Claim 12, wherein each cell of the submap has an associated signed distance field (SDF), and wherein the corresponding cell of the submap is identified as a cell having a disappearing object
if its SDF is higher than a second threshold.

15. The system according to Claim 11, wherein obtaining a submap of an area in the environment comprises:

obtaining the submap of the area in the environment based on sensor data provided by a robot deployed to move along a path in the environment during an operation.

16. The system according to Claim 15, wherein updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is carried out if one of the number of cells and the percentage of cells exceeds a change threshold.

17. The system according to Claim 16, wherein the processor is further configured to:

obtain a later version of the submap of the area in the environment;
determine cells of the static map corresponding to the cells of the later version of the submap whose occupancy status has changed;
obtain a second percentage of cells of the static map whose occupancy status has changed; and
apply Bayesian filtering to the first and the second percentage to obtain an updated percentage;

wherein updating occupancy status of the cells of the static map with occupancy status of the cells of the submap comprises updating occupancy status of the cells of the static map with occupancy status of the cells of the submap if the updated percentage exceeds a change threshold.

18. The system according to Claim 15, wherein obtaining the submap of the area in the environment based on sensor data provided by the robot deployed to move along the path in the environment during the operation comprises:

obtaining multiple scan groups over a time interval as the robot move along the path;

processing the multiple scan groups to identify any dynamic object in the area; and

generating the submap that excludes any identified dynamic object.

19. The system according to Claim 11, wherein the percentage of cells is a first percentage and a later version of the submap is generated based on sensor data from the robot as it moves along the path resulting in a second percentage of cells of the static map whose occupancy status has changed, the second percentage being different from the first percentage, wherein the processor is further configured to apply Bayesian filtering to the first and the second percentage to obtain an updated percentage, and updating occupancy status of the cells of the static map with occupancy status of the cells of the submap if the updated percentage exceeds the change threshold. [Claim 20] The system according to Claim 11, wherein updating occupancy status of the cells of the static map with the occupancy status of the cells of the submap is performed autonomously.

20. The system according to Claim 11, wherein the device comprises a robot.