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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 December 2016 (29.12.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/207669 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/GB2016/052592

(22) International Filing Date:
22 August 2016 (22.08.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1511065.3 23 June 2015 (23.06.2015) GB

(71) Applicant: **OXFORD UNIVERSITY INNOVATION LIMITED** [GB/GB]; Buxton Court, 3 West Way, Botley, Oxford Oxfordshire OX2 0JB (GB).

(72) Inventors: **NEWMAN, Paul**; c/o Oxford University Innovation Limited, Buxton Court, 3 West Way, Botley, Oxford Oxfordshire OX2 0JB (GB). **PINIES, Pedro**; c/o Oxford University Innovation Limited, Buxton Court, 3 West Way, Botley, Oxford Oxfordshire OX2 0JB (GB). **PAZ, Lina Maria**; c/o Oxford University Innovation Limited, Buxton Court, 3 West Way, Botley, Oxford Oxfordshire OX2 0JB (GB). **TANNER, Michael**; c/o Oxford University Innovation Limited, Buxton Court, 3 West Way, Botley, Oxford Oxfordshire OX2 0JB (GB).

(74) Agent: **BARKER BRETTELL LLP**; 100 Hagley Road, Edgbaston, Birmingham, West Midlands B16 8QQ (GB).

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

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

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

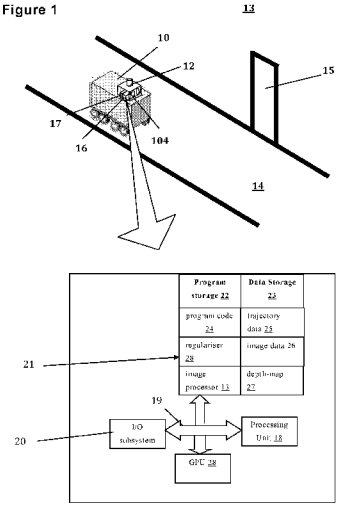
Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

— *the filing date of the international application is within two months from the date of expiration of the priority period (Rule 26bis.3)*

(54) Title: A METHOD OF GENERATING A THREE DIMENSIONAL REPRESENTATION OF AN ENVIRONMENT OR SYSTEM

Figure 1



(57) Abstract: A method of generating a three-dimensional (3D) representation of an environment or system, wherein the method comprises the following steps, i) obtaining a plurality of sets of range-data each providing range-data to features within at least a portion of the environment or system; ii) processing the plurality of sets of range-data to fuse the data and generate a set voxels holding data to represent the 3D environment or system; ii) identifying one or more subsets of the set of voxels;and iii) applying a regularisation process to at least one of the one or more subsets to modify the data held by at least some of the voxels in the one or more subsets wherein the set of voxels provides the 3D representation of the environment.

WO 2016/207669 A2

A METHOD OF GENERATING A THREE DIMENSIONAL REPRESENTATION OF AN ENVIRONMENT OR SYSTEM

Embodiments of the invention relate to a method of generating a three dimensional representation of an environment or system. In particular, but not exclusively, the invention may be used to generate three-dimensional (3D) reconstructions of environments by fusing depth-maps, or other range-data. Further, and in particular, embodiments may generate 3D representations of an environment through which a vehicle is travelling which the skilled person may refer to as large scale workspace mapping.

It is convenient to describe the background in terms of generating a 3D model of an environment around one or more vehicles, robots, or the like. However, the skilled person will appreciate that embodiments of the invention have wider applicability.

Building maps and workspace acquisition are established and desired competencies in mobile robotics. Improving the quality of maps of an environment will typically lead to better operation within that environment and workspace understanding. An important thread of work in this area is dense mapping in which, in stark contrast to the earliest sparse-point feature maps in mobile robotics, the goal is to construct continuous surfaces. This is a well-studied and vibrant area of research.

A precursor to many dense reconstruction techniques are 2.5D depth maps. These can be generated using a variety of techniques: directly with RGB-D cameras, indirectly with stereo cameras, or from a single camera undergoing known motion, and the like.

RGB-D (range finding RGB cameras) cameras are cameras which record depth information in addition to colour information for each pixel (Red, Blue, Green and Depth). However, RGB-D cameras are inappropriate for use over large scales, which may be considered to be more than roughly 5 meters, and outdoors, leading to inaccurate 3D representations.

RGB-D sensor-driven work often uses Microsoft Kinect or Asus Xtion PRO devices for example. Such RGB-D systems provide Video Graphics Array (VGA) colour and depth images at around 30 Hz, but this is at the cost of range (0.8 m to 3.5 m) and the ability to only reliably operate indoors (see, for example, Xtion PRO - specifications. http://www.asus.com/uk/Multimedia/Xtion_PRO/specifications), although outdoor operation is possible at night and with the same range limitation (see Whelan, T., Kaess, M., Fallon, M.F., Johannsson, H., Leonard, J.J., McDonald, J.B.: Kintinuous, "Spatially extended KinectFusion", RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras. Sydney, Australia (2012)). However, for the

indoor environments these structured light sensors can operate in, they produce accurate 3D dense reconstructions even in low-texture environments.

RGB-D based reconstructions rely on high quality depth maps always being available. In this case, regularisation may not be required since an average of the depth measurements can provide visually appealing results. Here regularisation may be better thought of as processing to produce locally smooth surfaces.

By contrast, when using camera-derived depth-maps, it is noted that the generated depth maps are almost always noisy and ill-formed in places - particularly a problem when operating in regions where there is a dearth of texture. Accordingly, regularisation processes may be applied to reduce these effects - essentially introducing a prior over the local structure of the workspace (planar, affine, smooth, etc). One such example is described in GB Patent Application GB1507013.9 which is hereby incorporated by reference and the skilled person is directed to read this application.

Stereo cameras also enable dense reconstruction, but introduce complexity and concerns around stable extrinsic calibration to the degree that they can be cost-prohibitive for low-end robotics applications (see Bumblebee2 FireWire stereo vision camera systems, Point Grey cameras, http://www.ptgrey.com/bumblebee2-firewire-stereo-vision-camera-systems).

An alternative approach is to leverage a sequence of mono images. In this case, there may be a need for an external method to derive, or at least seed, accurate estimates of the inter-frame motion of the camera - perhaps from an inertial measurement unit (IMU)-aided Visual Odometry system or a forward kinematic model of an arm. Using sets of sequential camera images with known inter-frame position and angle changes, 3D reconstructions of an outdoor or large-scale environment can be built up. However, the depth maps produced by such techniques are notoriously noisy. Extensive regularisation is therefore often used, which is computationally intensive and often inaccurate.

Embodiments described herein describe how data from depth maps are recombined and so it is assumed that inter-frame motion estimating methods are known to the skilled person. However the reader is pointed to Li, M., Mourikis, A.I., “High-precision, consistent EKF-based visual–inertial odometry”, The International Journal of Robotics Research 32(6), 690–711 (2013) for an example system. With the pose estimates between sequential images as a given, the depth of each pixel can be estimated using an identical approach to that taken in creating depth maps from stereo cameras (see, for example, Geiger, A., Roser, M., Urtasun, R., “Efficient large-scale

stereo matching", Asian Conference on Computer Vision (ACCV) (2010) or Hirschmuller, H., "Semi-global matching-motivation, developments and applications", hgpu.org (2011))

Full 3D dense reconstruction has been demonstrated in either indoor environments (see Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M., Bathiche, S., "MonoFusion: Realtime 3D reconstruction of small scenes with a single web camera", 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 83–88 (2013)) or small-scale outdoor environments (see Zach, C., Pock, T., Bischof, H., "A globally optimal algorithm for robust TV-L 1 range image integration", Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference (2007) or Graber, G., Pock, T., Bischof, H., "Online 3D reconstruction using convex optimization", 1st Workshop on Live Dense Reconstruction From Moving Cameras, ICCV 2011). Both of these methods rely on a fully-observed environment in which the observer orbits the subject which may be thought of as being object-centred in situ.

Thus, prior art techniques tend to be object-centred in situ, where the camera trajectory is chosen to generate high quality depth maps. In many mobile robotics applications - eg, an autonomous vehicle limited to an on-road trajectory - the environment observations are constrained and suboptimal for these traditional dense reconstruction techniques.

Such an object-centred in situ approach is exemplified in Figure 3a. As such, the surface of an object/environment to be reconstructed is seen by the sensors and there is at least an implicit assumption, that the observed voxels constitute the full set of voxels (ie $\Lambda = \Omega$ *using the notation found below)*. Inside the object in question, there may be lines and planes generated by erroneous interpolation and extrapolation to a region which the sensors cannot penetrate. In more general cases, such as in embodiments being described, sensors may move within an environment or workspace of which a representation is to be generated, as shown in Figure 3b. Prior art techniques are therefore ill-adapted to generating accurate representations in such circumstances, as portions of the workspace may not be visible, and data may be erroneously interpolated or extrapolated to fill the hidden space, thus the faithfulness of any representation of the environment that is so generated, is compromised.

Thus, embodiments address at least some of the deficiencies of current approaches to building 3D representations.

According to a first aspect of the invention there is provided a method of generating a three-dimensional (3D) representation of an environment or system, wherein the method comprises at least one of the following steps:

i) processing at least one representation of the environment or system to generate a set of voxels;

ii) identifying one or more subsets of the set of voxels;

iii) applying a regularisation process to at least one of the one or more subsets; and

iv) generating a 3D representation of the system or environment from the processed set of voxels.

Embodiments providing such a method are advantageous through the improved representations of the environment and/or system that are generated thereby.

Some embodiments may employ a different regularisation process for each subset, and may leave at least one sub-set unregularised.

According to a second aspect of the invention there is provided a processing apparatus, comprising at least one processor programmed to perform at least one of the following steps to generate a three-dimensional (3D) representation of an environment or system:

i) process at least one representation of the environment or system to generate a set of voxels;

ii) identify one or more subsets of the set of voxels;

iii) apply a regularisation process to at least one of the one or more subsets; and

iv) generate a 3D representation of the system or environment from the processed set of voxels.

According to a third aspect of the invention there is provided a machine readable medium containing instructions which when read by a machine cause at least one processor perform at least one of the following steps to generate a three-dimensional (3D) representation of an environment or system:

i) process at least one representation of the environment or system to generate a set of voxels;

ii) identify one or more subsets of the set of voxels;

iii) apply a regularisation process to at least one of the one or more subsets; and

iv) generate a 3D representation of the system or environment from the processed set of voxels.

The machine readable medium referred to in any of the above aspects of the invention may be any of the following: a CDROM; a DVD ROM / RAM (including -R/-RW or +R/+RW); a hard drive; a memory (including a USB drive; an XQD card, an SD card; a compact flash card or the like); a transmitted signal (including an Internet download, ftp file transfer of the like); a wire; etc.

Features described in relation to any of the above aspects of the invention may be applied, mutatis mutandis, to any of the other aspects of the invention.

There now follows by way of example only a detailed description of embodiments of the invention with reference to the accompanying drawings in which:

Figure 1 is a schematic view of a robot utilising a camera to take and process images of an environment in accordance with an embodiment;

Figure 2a is a graphical depiction of how Truncated Signed Distance Function values represent a surface in a two-dimensional voxel grid;

Figure 2b is a graphical depiction of how the Truncated Signed Distance Function values are discretised into histogram bins;

Figure 3a (Prior Art) is a graphical depiction of prior art object-centred applications of voxel-grid-based reconstruction;

Figure 3b is a graphical depiction of an environment traversed by a robot for an environmental application of voxel-grid-based reconstruction as described herein;

Figure 4 is a comparison of a prior art method and a method of an embodiment when applied to 3D reconstruction of a synthetic environment;

Figure 5 is a comparison of a prior art method and a method of an embodiment when applied to 3D reconstruction of a real-word, outdoor environment;

Figure 6 is a comparison of a prior art method and a method of an embodiment when applied to 3D reconstruction of a real-word, indoor environment;

Figure 7 is a flow chart illustrating the method steps of an embodiment;

Figure 8a shows continuous, dense reconstructions of an indoor environment, obtained from an embodiment and which may be thought of as a 3D representation of that environment; and

Figure 8b shows continuous, dense reconstructions of an outdoor environment and an outdoor environment, obtained from an embodiment and which may be thought of as a 3D representation of that environment; and

The claimed invention is described in relation to an embodiment having a sensor 12 mounted upon a robot 10. However, the skilled person will understand that other embodiments may not have this arrangement and for instance, the robot 10 could be replaced by a manned vehicle, or by a person carrying a sensor 12, or by a machine modelling a system, amongst other options. However, returning to the embodiment being described, the sensor 12 is arranged to monitor its environment 14, 15 and generate data based upon the monitoring, thereby providing data on a sensed scene around the robot 10 which is sometimes referred to as large scale workspace mapping. Thus, because the sensor 12 is mounted upon a robot 10, the sensor 12 is also arranged to monitor the environment 14, 15 of the robot 10.

Other embodiments may be used to generate a representation of a system rather than an environment. Here a system may be thought of as being a representation of a physical system such as a gas model, a rocket plume, or the like, where data can be fused from multiple representations of that system.

In the embodiment being described, the sensor 12 is a passive sensor (ie it does not create radiation and merely detects radiation) such as a camera. In the embodiment being described, the sensor 12 is a monocular camera.

The skilled person will appreciate that other kinds of sensor 12 could be used. In other embodiments, the sensor 12 may comprise other forms of sensor such as a laser scanner (such as a LIDAR, light detection and ranging, scanner) or the like. As such, the sensor 12 may also be an active sensor arranged to send radiation out therefrom and detect reflected radiation.

In the embodiment shown in Figure 1a, the robot 10 is travelling along a corridor 14 within a building 13 and the sensor 12 is imaging the environment (eg the corridor 14, door 15, etc.) as the robot 10 moves. The skilled person would understand that the robot may be remotely

controlled, may be following a pre-programmed route, or may calculate its own route, or any combination of these or the like.

In the embodiment being described, the robot 10 comprises processing circuitry 16 arranged to capture data from the sensor 12 and subsequently to process the data (in this embodiment, these data comprise images) generated by the sensor 12. Embodiments of the invention are described in relation to generating 3D representations of the environment around the sensor from RGB images 100 taken from a moving sensor 12. The skilled person would understand that other image types may be used, that a camera 12 taking the images 100 may not be in motion, and that multiple cameras and/or robots or the like may be used, wherein each robot may take a different route through the building 13. Further, the skilled person would understand that other forms of data may be used in the place of images – for example LIDAR point clouds.

As described hereinafter, colour taken from the image (here an RGB image) may be used as a soft segmentation cue. Here a soft segmentation cue may be thought of as being secondary information about a pixel in addition to the positional information provided by the pixel. In alternative embodiments, in which representations of the environment are used other than images, other soft segmentation cues may be used. For example, reflectance may be used.

Herein, the term depth-map is intended to mean a record of the distance of the surfaces of objects within the environment observed by the sensor 12 from a reference associated with the sensor 12. The reference may be a point reference, such as a point based on the sensor 12, or may be a reference plane. The distance to the surface may be recorded in any suitable manner. A depth-map is an example of range-data; ie data that gives information on range. Other examples of range-data may be the output of Laser scans (for example LIDAR scans)

Surfaces are an example of features within the environment. In addition to surfaces, an environment may well comprise edges, vertices, and the like.

In some embodiments, the distance to the surface may be recorded as a single value, associated with a pixel of an image 100. The image 100 may be thought of as providing an x-y plane. In one embodiment, the distance value associated with (or provided by) a pixel of an image may provide a depth value, and may be thought of as a z-value. Thus, the pixel with associated distance may therefore be thought of as range-data

Thus, the processing circuitry 16 captures data from the sensor 12, which data provides an image, or other representation, of the environment around the robot 10 at a current time. In the embodiment being described, the processing circuitry 16 also comprises, or has access to, a

storage device 17 on the robot 10. As such, the embodiment being described may be thought of as generating 3D-representations of an environment on-line. Here online means in what may be termed in real-time as the robot 10 moves within its environment 14, 15. As such, in real time might mean that the processing circuitry is able to process images at substantially any of the following frequencies: 0.1Hz, 0.5Hz, 1Hz; 2Hz; 5Hz; 10Hz; 15Hz; 20Hz; 25Hz; 30Hz; 50Hz (or any frequency in-between these). The skilled person would understand that the speed of data processing is limited by the hardware available, and would increase with hardware improvements.

The lower portion of Figure 1a shows components that may be found in a typical processing circuitry 16. A processor 18 may be provided which may be an Intel® X86 processor such as an i5, i7 processor, an AMDTM PhenomTM, OpteronTM, etc, an Apple A7, A8, A9 or A10 processor, or the like. The processor 18 is arranged to communicate, via a system bus 19, with an I/O subsystem 20 (and thereby with external networks, displays, and the like) and a memory 21.

The processor 18 also has access to a Graphics Processing Unit (GPU) 28 which is arranged to process large amounts of data in parallel so that embodiments using such a GPU 28 can be arranged to process data from the sensor 12 more rapidly than other embodiments. The GPU may be part of a CUDA (Compute Unified Device Architecture) platform.

The skilled person will appreciate that memory 21 may be provided by a variety of components including a volatile memory, a hard drive, a non-volatile memory, etc. Indeed, the memory 21 may comprise a plurality of components under the control of, or at least accessible by, the processor 18.

However, typically the memory 21 provides a program storage portion 22 arranged to store program code 24 which when executed performs an action and a data storage portion 23 which can be used to store data either temporarily and/or permanently. The data storage portion stores image data 26 generated by the sensor 12 (or data for other representations).

Trajectory data 25 may also be stored; trajectory data 25 may comprise data concerning a pre-programmed route and/or odometry data concerning the route taken – for example data concerning movement of the wheels, data from an INS system (Inertial Navigation System), or the like.

In other embodiments at least a portion of the processing circuitry 16 and/or the storage device 17 may be provided remotely from the robot 10. As such, it is conceivable that processing of the

data generated by the sensor 12 is performed off the robot 10 or partially on and partially off the robot 10. In embodiments in which the processing circuitry is provided both on and off the robot then a network connection (such as a 3G (eg UMTS - Universal Mobile Telecommunication System), 4G (LTE - Long Term Evolution) or WiFi (IEEE 802.11) or like) may be used.

It is convenient to refer to a robot 10 travelling along a corridor 14 but the skilled person will appreciate that embodiments need not be limited to any particular mobile apparatus or environment. Likewise, it is convenient in the following description to refer to image data 100 generated by a camera 12 but other embodiments may generate and use other types of data.

The sensor 12, together with the processing circuitry 16 to which the sensor 12 is connected, and with the software running on the processing circuitry 16, form a system capable of producing representations of the environment 14, 15 around the sensor 12 from the images 100 collected. In the embodiment being described, the representations take the form of depth-maps, but other embodiments may generate other forms of range-data.

As the sensor 12/robot 10 moves, a set of images is generated and the data providing the images is input to the processing circuitry 16. Typically, parallax between consecutive images 100, together with the trajectory data 25, is used to generate depth estimates for points within the images 100. Each point may correspond to a pixel of any one of the images. The depth estimate information for each pixel forms a depth-map of the environment 14, 15. Each, or at least the majority, of the depth-maps may be stored in the data storage portion 23 as depth map data 27. Depth-maps may be thought of as 2.5-dimensional representations of the environment.

The at least one representation of the environment 14, 15 generated by the sensor 12 and processing circuitry 16 can then be further processed, as described herein, to generate a 3D representation of the environment 14, 15. In the embodiments being described, a number of representations of the environment are fused to generate a set of voxels, where the voxels form a 3D grid with each voxel representing an element in that grid. Thus, each voxel provides positional information regarding the 3D environment. The number of representations to be fused may be on the order of 100. However, the skilled person will appreciate that this is given as an example, and any number of representations might be fused. Other embodiments may fuse roughly any of the following: 10, 20, 30, 50, 70, 90, 110, 150, 500, 1000, or more.

Thus, in the embodiment being described, fusion of the range-data is accomplished by building and managing a cube model of voxels (ie a set of voxels), where the 3D space being mapped is represented as the cube model. The skilled person would understand that volumes of voxels of other shapes may be used instead of or as well as cubes. Advantageously, use of a cube model

of voxels may simplify the mathematics required. Further, the skilled person would understand that, in some embodiments, the representations of the environment used may not be depth-maps, or may comprise other formats in addition to depth-maps, for example point clouds and/or the like, or other forms of range-data.

Prior art techniques can interpolate lines or planes within objects and such interpolated lines or planes may exist although they were not observed by the sensor (ie they are unobserved), and/or because these lines or planes could be within a solid object (ie they are unobservable) therefore meaningless data is generated. A voxel may be classed as "observed" if at least one data point from at least one of the representations of the environment or system being used falls within that voxel. A voxel may be classed as "observable" if, either it has been classed as "observed", or if interpolation between data points suggests that data for that voxel could be collected. For example, if the data collected indicate a planar surface (eg a wall or the like), points on that planar surface for which no data were collected may be classed as observable. By contrast, points behind that surface (potentially within the wall, the other side of a wall, or the like) may be classed as unobservable. Thus, in this embodiment whether or not a voxel is classed as observable is used to identify a subset of the voxels.

The skilled person would understand that, in other examples, the choice of variable to generate a subset of voxels may be different. Alternatively, or additionally, the choice of variable processes may be based on other context information. For example, image recognition or other processing may be used to identify certain objects (eg cars, people, trees, pipes, planar surfaces, or the like), and a different regularisation process may then be used for voxels corresponding to those objects, as compared to the process used for the remaining voxels.

Data (which may be thought of as context information) may be stored, or otherwise associated, with voxels within the grid of voxels. Embodiments may use such data stored, or otherwise associated, with a voxel to generate the sub-set of voxels.

Given a set of noisy dense depth maps from a sub set of monocular images, fusion of those depth maps is, in the embodiment being described, formulated as a regularised energy minimisation problem acting on the Truncated Signed Distance Function (TSDF) that parametrises the surface induced by the fusion of multiple depth maps. The solution is represented as the zero-crossing level of a regularised cube. In the embodiment being described, the optimisation and regularisation is executed in a 3D volume which has been only partially observed while avoiding inappropriate interpolation and extrapolation.

In other embodiments different variables are used to constrain the subset of voxels input to the 3D cube model and thus subsequently constrain the optimisation and regularisation.

The signed distance function (SDF) of a set, S, in a metric space determines the distance of a given point, x, from the boundary of S. The sign of the function is determined by whether or not x is within S. The function has positive values at points x inside S, it decreases in value as x approaches the boundary of S, is zero at the boundary of S, and takes negative values outside of S. The negative values become more negative (larger modulus) further outside the boundary of S. The skilled person will understand that positive values being inside S and negative values outside S is a matter of convention, and that the opposite convention can be applied when it is suitable, as is the case herein.

In the embodiments being described, only part of the distance data is needed to represent the surface of the object(s)/environment; the distance can therefore be truncated – a truncated signed distance function is therefore used. As calculated SDFs are only approximations of the true distance function, they can be erroneous, especially when estimated distances are large. However, for reconstruction, the voxels at or near the surface are of most importance when reconstructing the surface; in particular, it is important that a small band around the zero-crossing is accurately estimated in the SDF. Therefore, the projected distances are truncated.

The embodiments being described concern a technique that achieves 3D dense reconstruction with monocular cameras, with an input range from roughly 1.0 m to roughly 75 m. The embodiment being described also performs in regions of low texture which provides a low amount of information for the fusion of the depth maps and does not require privileged camera motion.

Embodiments may be used either or in both indoors and outdoors, and empirical analysis of the precision of the reconstructions is provided below.

The cube model is a discretised version of a Truncated Signed Distance Function (TSDF) $u : \Omega \rightarrow \mathbb{R}$ where $\Omega \subset \mathbb{R}^3$ represents a subset of points in 3D space and u returns the corresponding truncated distance to surfaces in the scene (see, for example, Curless, B., Levoy, M., "A volumetric method for building complex models from range images", Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 303–312. ACM (1996)). The TSDF is constructed in such a way that zero is the surface of an object, positive values represent empty space, and negative values correspond to the interior of objects, as shown in Figure 2. Thus by finding the zero-crossing level-set, $u = 0$, a dense representation of surfaces in the workspace can be achieved. Figure 2a shows a graphical depiction 200 of how

the TSDF values represent the zero-crossing surface in a two-dimensional voxel grid 202. Line 204 represents the surface observed by the camera 12. As can be seen in Figure 2a, values in the voxel grid 202 between the camera 12 and the surface 204 are positive. Values in the voxel grid 202 behind the surface 204 (from the point of view of the camera 12) have negative values. These values are TSDF values. As mentioned previously, only measurements near the surface 204 are of interest – the other voxels represent empty space or unobserved space. Distances are therefore truncated around the surface 204.

In Figure 2b, these TSDF values are discretised into histogram bins 250 (n_{bins} =5). $u \in [-1,1]$, which directly maps into histogram bins with indices from 1 to n_{bins}. There is no u value and no histogram bin when $u \leq -\mu$ (truncation of the distance behind the surface 204), however the n_{bins} histogram bin includes all $u > \mu$. Here μ is an absolute value that provides a metric for the confidence region for the surface observations. Highly accurate surface observations will have a small μ value, while inaccurate measurements will have a large μ value.

Consider first the case of operating with a single depth map D, an image in which each pixel (i, j) represents the depth $d_{i,j}$ of the closest obstacle in space along the z-axis. The 4×4 homogeneous matrix $\mathbf{T}_{gc} \in SE(3)$ is used to express the depth map's camera position, c, with respect to the voxel grid's global frame, g.

For each voxel, the steps to obtain u from a single depth map D are as follows:

1. Calculate the central point $\boldsymbol{p}_g = [x_g, y_g, z_g]^T$ of the voxel with respect to the camera coordinate frame as $\boldsymbol{p}_c = \boldsymbol{T}_{gc}^{-1}\boldsymbol{p}_g$;

2. Compute the pixel (i, j) in D in which the voxel is observed by projecting $\boldsymbol{p}_c$ into D and rounding each index to the nearest integer;

3. If the pixel (i, j) lies within the depth image, evaluate u as the difference between $d_{i,j}$ and the z component of $\boldsymbol{p}_c$. If $u > 0$, the voxel is between the surface and the camera whereas $u < 0$ indicates the surface occludes the camera's view of the voxel; and

4. Finally, linearly scale-and-clamp u such that any voxel for which $u > -\mu$ lies in the interval $[-1,1]$ whereas voxels for which $u < -\mu$ are left empty

Embodiments also fuse multiple depth maps D_t obtained at different moments in time t which is now explained.

When high-quality depth maps are available, for example depth maps obtained from a 2.5D camera such as the Microsoft^{RTM} Kinect^{TM} camera, data fusion can be performed by minimising, for each voxel, the following L_2 norm energy,

$$\arg\min_u \int_\Omega \sum_{t=1}^{N} \|u - f_t\|_2^2 \, d\Omega \qquad \text{..... (Eq. 1)}$$

where N represents the number of depth maps we want to fuse, f_t is the TSDF that corresponds to depth map D_t and u is the optimised TSDF after fusing all the information available. Using a voxel grid representation for the TSDFs, the solution to this problem can be obtained by calculating the mean of all the $f_1, \ldots, f_N$ for each individual voxel. This operation can be performed in substantially real time, which is as discussed above, by sequentially integrating a new f_t when a new depth map is available (see Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A., "KinectFusion: Real-time dense surface mapping and tracking", Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium, pp. 127–136. IEEE (2011)). The searched TSDF u does not require any additional regularisation due to the high-quality of the depth maps used in the fusion.

However, when cameras are used, the depth maps obtained are of lower quality when compared to those obtained from a 2.5D camera due, for example, to poor parallax or incorrect pixel matches. Therefore a more robust method is used. In the paper of Zach, C., Pock, T., Bischof, H. cited previously, the authors propose an L_1 norm data term, which is able to cope with spurious measurements, and an additional regularisation term, based on Total Variation (see Rudin, L.I., Osher, S., Fatemi, E., "Nonlinear total variation based noise removal algorithms", Proc. of the 11th annual Int. Conf. of the Center for Nonlinear Studies on Experimental mathematics: computational issues in nonlinear science, pp. 259–268. Elsevier North-Holland, Inc.(1992)), to smooth the surfaces obtained. The energy minimised is given by,

$$\arg\min_u \int_\Omega |\nabla u|_1 + \lambda \int_\Omega \sum_{t=1}^{N} \|u - f_t\|_1 \, d\Omega \qquad \text{..... (Eq. 2)}$$

The first component is a smoothness term that penalises high-varying surfaces, while the second component, which mirrors Eq. 1, substitutes the L_2 norm with a robust L_1 energy term. The parameter $\lambda > 0$ is a weight to trade-off between the regularisation and the data terms. The main drawback with this approach is that, unlike the fusion of depth maps obtained from a 2.5D camera, the TSDF u cannot simply be sequentially updated when a new depth map arrives,

instead, this method requires all previous history of depth values in each voxel to be stored. This limits the number of depth maps that can be integrated/fused due to memory requirements.

Thus, the embodiment being described overcomes this limitation, since by construction the TSDFs f_t integrated are bounded to the interval $[-1,1]$, Zach, C. proposes, in "Fast and high quality fusion of depth maps", Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT) (2008), sampling this interval by evenly spaced bin centres c_b (see Figure 2) and approximating the previous data fidelity term $\sum_{t=1}^{N}|u - f_t|_1$ by $\sum_{b=1}^{n_{bins}} h_b|u - c_b|_1$ where h_b is the number of times the interval has been observed. The corresponding energy for the histogram approach is,

$$\arg\min_u \int_\Omega |\nabla u|_1 + \lambda \int_\Omega \sum_{b=1}^{n_{bins}} h_b|u - c_b|_1 \, d\Omega \qquad \text{..... (Eq. 3)}$$

where the centre of the bins are calculated using,

$$c_b = \frac{2b}{n_{bins}} - 1 \qquad \text{..... (Eq. 4)}$$

The voting process in the histogram is depicted in Figure 2b. While this voting scheme, described in the paper of Zach, C. cited above, significantly reduces the memory requirements, allowing an unlimited number of depth maps to be integrated, the embodiment being described uses a further refinement described in Li, Y., Osher, S., et al., "A new median formula with applications to PDE based denoising", Commun. Math. Sci 7(3), 741–753 (2009) and has been applied to histogram-based voxel grids by Graber, G., Pock, T. and Bischof, H., in their 2011 paper cited above. Further details of the further refinement are described below after the Ω domain is introduced.

As mentioned previously, traditional voxel-grid-based reconstructions focus on object-centred applications 300 as depicted in Figure 3a (Prior Art). In this scenario, the objects 302 in the voxel grid are fully observed multiple times from a variety of angles, by one or more sensors 12. Even though the internal portion 304 of the object 302 has not been observed, previous regularisation processes do not make a distinction between Ω (observed regions, 302) and $\bar{\Omega}$ (unobserved regions, 304). This results in spurious interpolation inside the object 302. However, in mobile robotics applications 350 the world environment 352, 354 is traversed and observed during exploration, requiring large voxel grids, as shown in Figure 3b, which result in significant portions of the environment never being observed 354.

For example, at camera 12 capture t_x, it is unknown what exists in the camera's upper field of view. Not accounting for $\bar{\Omega}$ in regularisation results in incorrect surface generation. The embodiment being described defines Λ as the voxel grid domain while Ω is the subset of Λ which has been directly observed and which will be regularised.

Different domain intervals (Ω and $\bar{\Omega}$), which may also be thought of as different sub-sets of voxels in the voxel cube model, are therefore defined. This redefinition of the domain, Λ, allows regularisation and interpolation in places of interest within the environment.

Multiple surface observations, which may be obtained from one or more cameras, laser/LIDAR systems or other sensors, are fused into the 3D cube model of voxels. Once all input data is fused, the embodiment being described targets a subset, labelled the Ω domain, of the volume for regularisation. Embodiments may apply regularisation for a number of reasons and different regularisers may be applied to different sub-sets of voxels. For example embodiments may be arranged to smooth out noisy data, interpolate unobserved surfaces, use a geographic constraint to improve the appearance of objects, locate vehicles within the data, or the like.

For example, a sparse point cloud produced by a forward-moving vehicle can include the "empty" space between sequential laser scans in the Ω domain. This results in a continuously interpolated surface reconstruction wherein points are filled in to produce smooth surfaces.

In the embodiment being described, where the Ω domain is used to regularise voxels that have been observed as described in Figure 3b, an advantage is the processing prevents the creation of spurious surfaces during regularisation – the method recognises that no data are available for voxels in the $\bar{\Omega}$ (unobserved) set.

In the embodiments being described, the sensor(s) 12 are moving within the voxel grid and only observe a subset of the overall voxels. Thus, as is used in the embodiment being described in relation to Figure 3b a regulariser is used to prevent the unobserved voxels from negatively affecting the regularisation results of the observed voxels. In order to achieve this, as illustrated in Figure 3b, the complete voxel grid domain is defined as Λ, and Ω is used to represent the subset of voxels which have been directly observed and which, in the embodiment being described, will be regularised. The remaining subset, $\bar{\Omega}$, represents voxels which have not been observed in the data being processed. By definition, Ω and $\bar{\Omega}$, form a partition of Λ and therefore $\Lambda = \Omega \cup \bar{\Omega}$, and $\Omega \cap \bar{\Omega} = \emptyset$. Therefore, $\Omega \subset \Lambda$ as Figure 3b illustrates.

In this case Equation 3 becomes,

$$\arg\min_{u} \int_{\Lambda} |\nabla u|_1 + \lambda \int_{\Omega} \sum_{b=1}^{n_{bins}} h_b |u - c_b|_1 \, d\Omega \qquad \ldots\ldots\ldots \text{(Eq. 5)}$$

Note that $\bar{\Omega}$ voxels lack the data term. As is explained in Chambolle, A., Pock, T., "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging", Journal of Mathematical Imaging and Vision 40(1), 120–145 (2011), this regularisation technique interpolates the content of voxels in the subset of voxels denoted herein as $\bar{\Omega}$. Extrapolation occurs when there are unobserved voxels surrounding an observed region. To avoid this extrapolation, the embodiment being described sets the Ω domain boundary conditions to constrain regularisation to observed voxels, thus avoiding indiscriminate surface creation which would otherwise occur.

As described above, heterogeneous processing is therefore performed on the two subsets, Ω and $\bar{\Omega}$, identified within the domain Λ.

The skilled person would understand that the Ω-domain principles could be applied to new boundary conditions which select portions of the voxel grid for regularisation. These subsets could be selected based on scene-segmentation heuristics, such as context information. Such context information may be stored, or otherwise associated with, voxels within the set of voxels.

Context information may include one or more of the following:

i. whether or not data are present for a particular voxel (whether it was observed);
ii. whether or not data are present, combined with interpolation of data between representations and/or between data points, perhaps by ray-tracing, use of geometrical assumptions, or the like, to determine whether or not data for a particular voxel should be obtainable, eg if the voxel corresponds to a surface of an object rather than the interior of an object (whether it was observable);
iii. sensor type (eg camera or LIDAR);
iv. colour information;
v. texture information;
vi. one or more geometrical assumptions (eg that the environment will comprise planar surfaces (such as inside a building), that the environment will comprise circular, cylindrical surfaces (such as might be the case in a chemical plant), or the like);
vii. reflectance information, which is advantageous for LIDAR data;
viii. labels or other metadata; and
ix. image recognition data (eg identifying cars or people).

To use a simplistic example, in a forest scene, images could be segmented by colour – for example, brown, green, blue. The method may then identify brown with the ground, green with foliage and blue with the sky, and interpolate and extrapolate between voxels with matching colour information as is deemed to be appropriate. The skilled person would understand that the Ω domain may be divided into more than two subsets where appropriate, where each subset is subsequently treated independently. In the example being given, three subsets would be used: a first for those voxels deemed 'brown'; a second for those voxels deemed 'green'; and a third for those deemed 'blue'.

By way of further example, the Ω domain can be extended to include enclosed "holes" which will result in the regulariser interpolating a new surface. Alternatively, a segment from Ω could be removed to prevent regularisation of a scene segment which was better estimated in the depth map (for example a high-texture object).

Irrespective of the information used to split the Ω domain into subsets, the method disclosed herein allows different regularisation processes to be applied to the different subsets of voxels, so facilitating more accurate interpolation and extrapolation, and so a more accurate 3D reconstruction.

The solution to Equation 3 above, is now described using the Ω-domain constraint outlined above as used in the embodiment being described and relating to whether or not the voxels have been observed. Note that both terms in Equation 3 are convex but not differentiable since they depend on the L_1 norm. To solve this, a Proximal Gradient method can be used, as described in the paper of Chambolle and Pock cited above, which requires transformation of one of the terms into a differentiable form. The Total Variation term is transformed using the Legendre-Fenchel Transform (see Rockafellar, R.T., "Convex Analysis", Princeton University Press, Princeton, New Jersey (1970)),

$$\min_u \int_\Omega |\nabla u|_1 d\Omega = \min_u \max_{\|\boldsymbol{p}\|_\infty \leq 1} \int_\Omega u \nabla . \boldsymbol{p} \, d\Omega \quad \text{..... (Eq. 6)}$$

where $\nabla . \boldsymbol{p}$ is the divergence of a vector field $\boldsymbol{p}$ defined by $\nabla . \boldsymbol{p} = \nabla p_x + \nabla p_y + \nabla p_z$.

Applying this transformation to Equation 3 the original energy minimisation problem turns into a saddle-point (min-max) problem that involves a new dual variable $\boldsymbol{p}$ and the original primal variable u,

$$\min_u \max_{\|\boldsymbol{p}\|_\infty \leq 1} \int_\Omega u \nabla . \boldsymbol{p} \, d\Omega + \lambda \int_\Omega \sum_{b=1}^{n_{bins}} h_b |u - c_b|_1 \, d\Omega \quad \text{..... (Eq. 7)}$$

The solution to this regularisation problem was demonstrated in the paper of Graber, Pock, and Bischof cited above, with a Primal-Dual optimisation algorithm (see the paper of Chambolle and Pock cited above) which is briefly summarised in the following steps:

1. $\boldsymbol{p}$, u, and $\bar{u}$ can be initialised to 0 since the problem is convex and is guaranteed to converge regardless of the initial seed. $\bar{u}$ is a temporary variable used to reduce the number of optimisation iterations required to converge;

2. To solve the maximisation, the dual variable $\boldsymbol{p}$ is updated,

$$\boldsymbol{p} = \boldsymbol{p} + \sigma\nabla\bar{u} \qquad \text{..... (Eq. 8)}$$

$$\boldsymbol{p} = \frac{\boldsymbol{p}}{max\,(1, \|\boldsymbol{p}\|_2)}$$

where σ is the dual variable gradient-ascent step size;

3. For the minimisation problem, the primal variable u is updated by,

$$u = u - \tau\nabla.\boldsymbol{p} \qquad \text{..... (Eq. 9)}$$

$$W_i = -\sum_{j=1}^{i} h_j + \sum_{j=i+1}^{n_{bins}} h_j \qquad i \in [0, n_{bins}]$$

$$b_i = u + \tau\lambda W_i$$

$$u = median(c_1, \ldots, c_{n_{bins}}, b_0, \ldots, b_{n_{bins}})$$

where τ is the gradient-descent step size, W_i is the optimal weight for histogram bin i, and b_i is the regularisation weight for histogram bin I;

4. Finally, to converge in fewer iterations, a "relaxation" step is applied,

$$\bar{u} = u + \theta(u - \bar{u}) \qquad \text{..... (Eq. 10)}$$

where θ is a parameter to adjust the relaxation step size.

The embodiment being described is arranged to compute equations 8, 9, and 10 for each voxel in each iteration of the optimisation loop. Since each voxel's computation is independent of that for every other voxel, this is implemented as a GPU 28 kernel which operates within the optimisation loop. The final output, u, represents the regularised TSDF distance.

Without loss of generality, the discrete gradient and divergence operations traditionally used to solve Equations 8 and 9 are described for the x component (see Chambolle, A., "An algorithm for total variation minimization and applications", Journal of Mathematical imaging and vision 20(1-2), 89–97 (2004)),

$$\nabla_x u_{i,j,k} = \begin{cases} u_{i+1,j,k} - u_{i,j,k} & if\ 1 \le i < V_x \\ 0 & if\ i = V_x \end{cases} \quad \text{..... (Eq. 11)}$$

$$\nabla_x \cdot \boldsymbol{p}_{i,j,k} = \begin{cases} \boldsymbol{p}^x_{i,j,k} - \boldsymbol{p}^x_{i-1,j,k} & if\ 1 < i < V_x \\ \boldsymbol{p}^x_{i,j,k} & if\ i = 1 \\ -\boldsymbol{p}^x_{i-1,j,k} & if\ i = V_x \end{cases} \quad \text{..... (Eq. 12)}$$

where V_x is the number of voxels in the x dimension. As would be understood by the skilled person, y and z components can be obtained by changing index i for j and k respectively.

The gradient and divergence calculations are extended to account for new conditions which remove the $\bar{\Omega}$ domain from regularisation. These methods can be intuitively thought of as introducing additional boundary conditions within the voxel cube which previously only existed on the edges of the voxel grid. For an input TSDF voxel grid u, the gradient $\nabla u = \left[\nabla_x u, \nabla_y u, \nabla_z u\right]^T$ is computed by Equation 11 with the following additional conditions,

$$\nabla_x u_{i,j,k} = \begin{cases} 0 & if\ u_{i,j,k} \in \bar{\Omega} \\ 0 & if\ u_{i+1,j,k} \in \bar{\Omega} \end{cases} \quad \text{..... (Eq. 13)}$$

Note that the regulariser uses the gradient to diffuse information among neighbouring voxels. The gradient definition provided herein therefore excludes $\bar{\Omega}$ voxels from regularisation.

Finally, in addition to the conditions in Equation 12, the divergence operator must be defined such that it mirrors the modified gradient operator:

$$\nabla_x \cdot \boldsymbol{p}_{i,j,k} = \begin{cases} 0 & if\ u_{i,j,k} \in \bar{\Omega} \\ \boldsymbol{p}^x_{i,j,k} & if\ u_{i-1,j,k} \in \bar{\Omega} \\ -\boldsymbol{p}^x_{i-1,j,k} & if\ u_{i+1,j,k} \in \bar{\Omega} \end{cases} \quad \text{..... (Eq. 14)}$$

To evaluate the performance of the technique of an embodiment, three experiments were performed comparing the cube model method outlined above to a KinectFusion implementation which fuses depth maps generated by a Microsoft™ Kinect™ camera. The dense reconstructions are executed on a NVIDIA GeForce GTX TITAN graphics card with 2,880 CUDA Cores and 6 GB of device memory.

As a proof of concept, a qualitative analysis of the algorithm was first undertaken on synthetic data (Figure 4) before performing more robust tests with real-world environments. The synthetic data set provides high-precision depth maps of indoor scenes taken at 30 Hz (see http://www.doc.ic.ac.uk/ahanda/VaFRIC/index.html, http://www.doc.ic.ac.uk/ahanda/High FrameRateTracking/downloads.html and also Handa, A., Whelan, T., McDonald, J., Davison, A., "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM", IEEE Intl. Conf. on Robotics and Automation, ICRA. Hong Kong, China (2014)). The chosen scene incorporates both close and far objects observed from the camera with partial occlusions. The input of the 3D reconstruction pipeline is a set of truth depth maps with added Gaussian noise (standard deviation, $\sigma_n = 10$ cm).

Figure 4 shows a comparison of the KinectFusion (left, A) and cube model regularisation (right, B) methods for a 3D reconstruction of a synthetic (see the paper of Handa, Whelan, McDonald, and Davison, cited above) environment by fusing noisy depth maps. The Phong shading shown in Figure 4 demonstrates how our regularisation produces consistent surface normals without unnecessarily adding or removing surfaces.

The skilled person would understand that Phong shading is an interpolation technique for surface shading in 3D computer graphics. It may also be referred to as normal-vector interpolation shading. More specifically, Phong shading interpolates surface normals across rasterised polygons and computes pixel colours based on the interpolated normals and a reflection model.

As can be seen in Figure 4, where results are represented using Phong shading, there is a significant improvement in surface normals when the scene is regularised with the cube model (Figure 4B) method, as compared to KinectFusion (Figure 4A).

A side-benefit of the regularised normals is that the scene can be represented with fewer vertices. It was found that the cube model scenes required 2 to 3 times fewer vertices than the same scene processed by KinectFusion.

To quantitatively analyse the cube model method, two real-world experiments were conducted in large-scale environments. Again, the cube model and KinectFusion fusion pipelines were compared, this time with depth maps generated from a monocular camera using the techniques described in Pinies, P., Paz, L.M., Newman, P., "Dense and Swift Mapping with Monocular Vision", International Conference on Field and Service Robotics (FSR). Toronto, ON, Canada (2015). The first (Figure 5) represents the 3D scene reconstruction of an urban outdoor environment in Woodstock, UK. The second (Figure 6) is a long, textureless indoor corridor of

the University of Oxford's Acland building. In both experiments, a frontal monocular camera was used, covering a field of view of 65°× 70° and with an image resolution of 512 × 384.

For ground truth, metrically consistent local 3D swathes were generated from a 2D push-broom laser using a subset of camera-to-world pose estimates $T_{WC} \in SE(3)$ in an active time window as $M_L = f(T_{WC}, T_{CL}, x_L)$, where f is a function of the total set of collected laser points x_L in the same time interval and T_{CL} is the extrinsic calibration between camera and laser. The resulting 3D point cloud M_L is used as ground truth for the large scale assessment.

Table 1 summarises the dimensions of the volume used for each of the experiments, the number of primal dual iterations, and the total running time required for the fusion approach. The execution time for regularisation is highly correlated to the size of the Ω space because regularisation is only performed on voxels within Ω. The timing results of cube model regularisation shown in Table 1 are for regularisation performed on an NVIDIA GeForce GTX TITAN graphics card. For the configuration parameters, only the volume's dimension changed, but the number of voxels (and hence memory requirements) remained consistent between experiments.

Table 1

Experiment	Voxels	Volume Dim (m)	Iterations	Regularisation time (s)	Memory size (MB)
Woodstock	512	6×25×10	100	11.09	640
Acland	512	4×6×30	100	11.24	640

Figures 5 and 6 show a comparison between the ground truth and the 3D reconstructions obtained using the cube model and the KinectFusion methods. To calculate the statistics, a "point-cloud-to-model" registration of the ground truth was performed with respect to the model estimate (see http://www.danielgm.net/cc).

Figure 5 is based on the Woodstock Data Set and shows a comparison of the KinectFusion (left, A) and cube model (right, B) dense reconstruction techniques. The KinectFusion has a larger number of spurious outlier segments and requires more than twice the number vertices to represent the structure due to its irregular surfaces.

In Figure 5 A and B, it can be seen that the KinectFusion implementation (,Figure 5A) produced a large range of spurious data points when compared to the cube model method, of the embodiment being described, (Figure 5B). The shaded vertices of Figures 5A and 5B correspond to the shading used in the histogram bins of Figures 5C and 5D. This spurious data is highlighted in the region 500 and it can be seen that the corresponding region 502 in Figure 5d has fewer returns.

Figure 5 C and D show histograms of per-vertex-error when compared to laser-generated point clouds for the data shown in Figures 5A and 5B. The KinectFusion (left, C) has a median error of 373 mm (σ = 571 mm) while the cube model (right, D) method has a median error of 144 mm (σ = 364 mm). Note that the cube model method requires fewer vertices to represent the same scene when compared to the KinectFusion implementation.

The cube model method's median and standard deviation are approximately half that of the KinectFusion method.

Figure 6 is the equivalent of Figure 5 for the Acland Data Set in place of the Woodstock Data Set. Figure 6 A and B show a comparison of the KinectFusion (Figure 6A) and cube model (Figure 6B) dense reconstruction techniques for the Acland Data Set. Note that the laser truth data was only measured depth data for the lower-half of the hallway. This results in the spurious errors for the upper-half where the depth maps produced estimates but for which there was no truth data. These errors dominate the right tail of the histograms in Figure 6 C and D.

In Figure 6 A and B, a comparison of Point Clouds is presented. The cube model (right, B) method again outperformed the KinectFusion implementation (left, A). The shaded vertices within Figures 6A and 6B correspond to the shading used in the histogram bins in Figure 6 C and D.

In Figure 6 C and D, histograms of per-vertex-error when compared to laser-generated point clouds are presented. The KinectFusion (left, C) has a median error of 310 mm (σ = 571 mm) while our cube model (right, D) method had a median error of 151 mm (σ = 354 mm). Note that the cube model method again requires fewer vertices to represent the same scene.

As with the Woodstock data set, the cube model method's median and standard deviation are approximately half that of the KinectFusion method.

The key statistics comparing the methods are precisely outlined in Table 2. Table 2 shows error analysis comparing KinectFusion and cube model methods. The cube model error is roughly

half that of KinectFusion. For both scenarios, the cube model method was therefore roughly two times more accurate than KinectFusion.

Table 2

Experiment	Median Error (m)	Standard Deviation (m)
Woodstock (KinectFusion)	0.3730	0.5708
Woodstock (cube model)	0.1441	0.3636
Acland (KinectFusion)	0.3102	0.5708
Acland (cube model)	0.1508	0.3537

Figure 7 is a flow chart illustrating the method steps 700 of an embodiment, as applied to the use of depth-maps to generate a representation of an environment.

At step 702, one or more depth-maps (sets of range data, or other representation of the environment as described above) are obtained. The depth-maps may be generated from an environment by any method. Each depth map comprises a plurality of points (or pixels) with depth estimates. The depth-maps may further comprise colour or texture information, or other information about the surface portrayed, for example labels to indicate that certain points correspond to “road” or “foliage”.

At step 704, the depth-maps are fused into a 3D volume of voxels. The skilled person would understand that many methods of fusing depth maps into a voxel grid are known, and that any of those methods may be employed. The texture information, or other information, may be stored, or otherwise associated with, the voxels.

At step 706, the voxels are split into two (or more) subsets. Figure 7 illustrates the process for two subsets, but the skilled person would understand that any number of subsets could be defined and treated accordingly. Dividing voxels into unobserved and observed subsets, as described above, is one example of splitting the voxels.

As illustrated by steps 708a and 708b, different regularisation processes may be appropriate to each subset. The first and second regularisation processes may be the same or different. In cases where more than two subsets are created, the number of different regularisation processes

used is smaller than or equal to X, where X is the number of subsets. It is noted that no regularisation may be applied to some of the sub-sets.

Once regularisation is complete, the voxels can be used to provide a 3D representation of the environment (step 712). However, the skilled person will appreciate that the voxels provide a 3D representation of the environment or system and this 3D representation may be utilised without being displayed or produced into tangible form; a machine, such as a computer, robot or the like, may find utility in processing the voxels held within the memory.

Advantageously, the approach described herein may allow regularisation to be applied to one or more subsets of the voxel grid, and that regularisation will neither modify nor be influenced by voxels outside of its subset.

The skilled person will appreciate that in the embodiment described herein, an input to the method are the images generated from the sensor 12. As discussed above, embodiments are arranged to process those images at real time as described above.

At least some embodiments may be arranged such that some of the processing described above is performed after multiple data inputs (images in the embodiment being described) have been fused into the system. Such embodiments may be advantageous in increasing the speed at which the processing can be performed and may be thought of as processing the data input to the system in batches.

Finally, Figure 8 shows the obtained continuous, dense reconstructions of the indoor and outdoor environments, which reconstructions may be thought of as 3D representations generated by the embodiment of the invention being described. More specifically, Figure 8 shows the final 3D reconstruction of the large scale experiments using cube model with the Acland building (top, A) and Woodstock, UK (below, B). Here, the generating step comprises displaying the 3D representation on a display. However, the skilled person will appreciate that the voxel grid provides a model of the 3D environment within the memory in which it is stored.

The skilled person will appreciate that embodiments described herein implement elements thereof as software. The skilled person will also appreciate that those elements may also be implemented in firmware or hardware. Thus, software, firmware and/or hardware elements may be interchangeable as will be appreciated by the skilled person.

CLAIMS

1. A method of generating a three-dimensional (3D) representation of an environment or system, wherein the method comprises the following steps:

i) obtaining a plurality of sets of range-data each providing range-data to features within at least a portion of the environment or system;

ii) processing the plurality of sets of range-data to fuse the data and generate a set of voxels holding data to represent the 3D environment or system;

ii) identifying one or more subsets of the set of voxels;

iii) applying a regularisation process to at least one of the one or more subsets to modify the data held by at least some of the voxels in the one or more subsets wherein the set of voxels provides the 3D representation of the environment.

2. The method of claim 1 which comprises a further step iv) of displaying or otherwise generating a 3D representation of the system or environment from the processed set of voxels.

3. The method of generating a 3D representation of an environment or system of claim 1 or 2, wherein a different regularisation process is used for each subset.

4. The method of generating a 3D representation of an environment or system of any preceding claim wherein at least one of the subsets is not regularised.

5. The method of generating a 3D representation of an environment or system of any preceding claim wherein the range data is provided by a depth-map.

6. The method of generating a 3D representation of an environment or system of any preceding claim, wherein the subsets of voxels are identified according to context information wherein the context information includes one or more of the following:

i. whether or not data are present;
ii. whether or not data can be interpolated;
iii. sensor type;
iv. colour information;
v. texture information;
vi. one or more geometrical assumptions;
vii. reflectance information;

viii. labels or other metadata;

ix. image recognition data.

7. The method of generating a 3D representation of an environment or system of any preceding claim wherein a Truncated Signed Distance Function (TSDF) is used to generate features within the range data.

8. The method of generating a 3D representation of an environment or system of claim 7 wherein the TSDF is processed to make it a discrete function.

9. A processing apparatus, comprising at least one processor programmed to perform the following steps to generate a three-dimensional (3D) representation of an environment or system:

i) obtain a plurality of sets of range-data each providing range-data to features within at least a portion of the environment or system;

ii) process the plurality of sets of range-data to fuse the data and generate a set of voxels holding data to represent the 3D environment or system;

iii) identify one or more subsets of the set of voxels;

iv) apply a regularisation process to at least one of the one or more subsets to modify the data held by at least some of the voxels in the one or more subsets wherein the set of voxels provides the 3D representation of the environment.

10. The apparatus of claim 9 which comprises arranging the processor to display or otherwise generate a 3D representation of the system or environment from the processed set of voxels.

11. The apparatus of claim 9 or 10 wherein the processor is programmed to apply a different regularisation process for each subset.

12. The apparatus of any of claims 9 to 11 wherein the processor is programmed such that at least one of the subsets is not regularised.

13. The apparatus of any of claims 9 to 12 wherein the processor is programmed such that the range data is a depth-map.

14. The apparatus of any of claims 9 to 13 wherein the processor is arranged such that the subsets of voxels are identified according to context information wherein the context information includes one or more of the following:

i. whether or not data are present;
ii. whether or not data can be interpolated;
iii. sensor type;
iv. colour information;
v. texture information;
vi. one or more geometrical assumptions;
vii. reflectance information;
viii. labels or other metadata;
ix. image recognition data.

15. The apparatus of any of claims 9 to 14 further comprising a monocular camera wherein the system is arranged to generate range data from images captured by the camera.

16. A machine readable medium containing instructions which when read by a machine cause at least one processor perform the following steps to generate a three-dimensional (3D) representation of an environment or system:

i) obtain a plurality of sets of range-data each providing range-data to features within at least a portion of the environment or system;

ii) process the plurality of sets of range-data to fuse the data and generate a set of voxels holding data to represent the 3D environment or system;

iii) identify one or more subsets of the set of voxels;

iv) apply a regularisation process to at least one of the one or more subsets to modify the data held by at least some of the voxels in the one or more subsets wherein the set of voxels provides the 3D representation of the environment.

17. The medium of claim 16 wherein the instructions are arranged to cause the processor to display or otherwise generate a 3D representation of the system or environment from the processed set of voxels.

18. The medium of claim 16 or 17 wherein the instructions are arranged to cause the processor to apply a different regularisation process for each subset.

19. The medium of any of claims 16 to 18 wherein the instructions are arranged to cause the processor substantially not to regularise at least one of the subsets.

20. The medium of any of claims 16 to 19 wherein the instructions are arranged such that the range-data comprises a depth-map

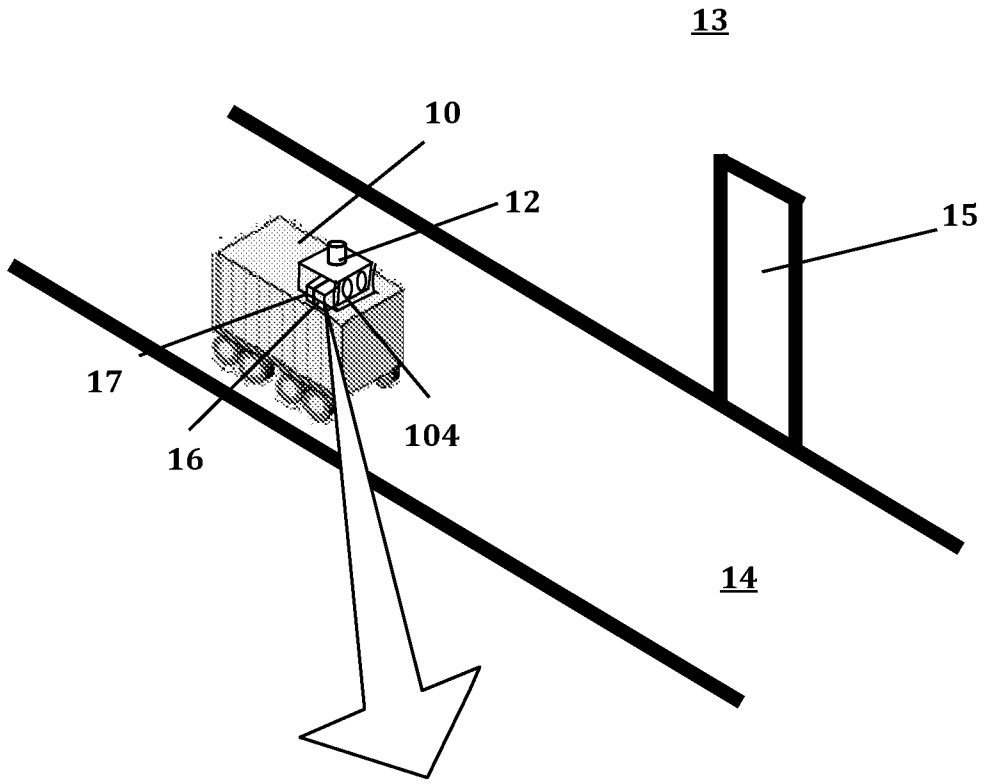
21. The medium of any of claims 16 to 20 wherein the instructions are arranged such that the subsets of voxels are identified according to context information wherein the context information includes one or more of the following:

i. whether or not data are present;
ii. whether or not data can be interpolated;
iii. sensor type;
iv. colour information;
v. texture information;
vi. one or more geometrical assumptions;
vii. reflectance information;
viii. labels or other metadata;
ix. image recognition data.

22. A robot or Autonomously Guided Vehicle (AGV) arranged to use the method of any of claims 1 to 8.

23. A robot or AGV according to claim 22 which comprises a self-driving car.

Figure 1


13
10
12
15
17
16
104
14

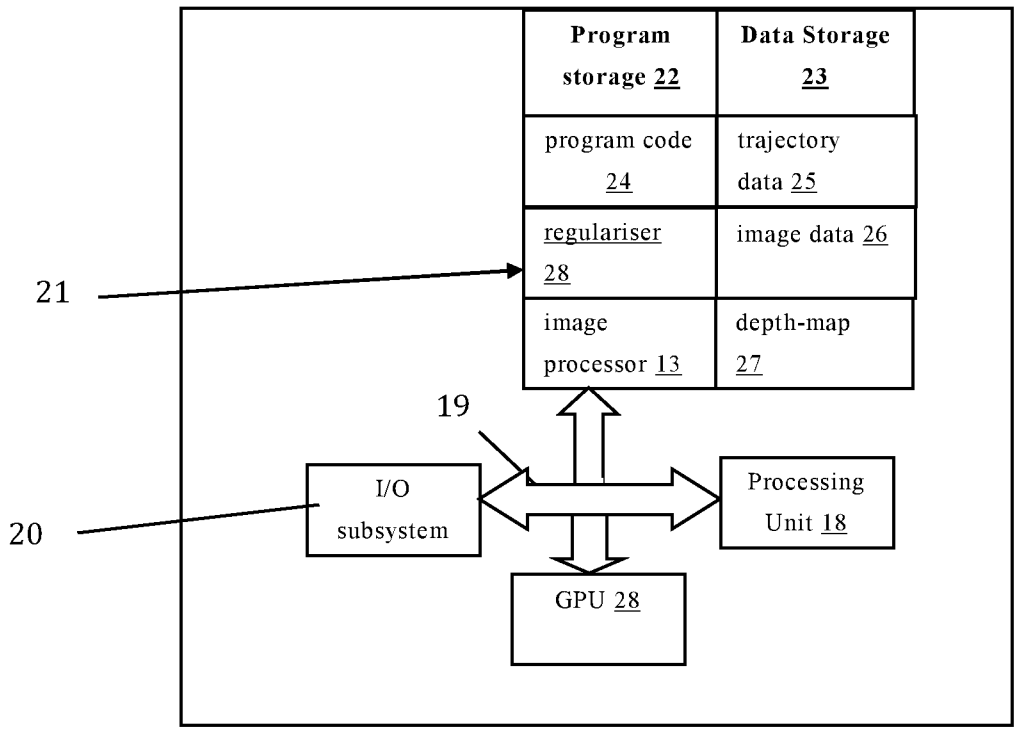

Program storage 22
Data Storage 23
program code 24
trajectory data 25
regulariser 28
image data 26
image processor 13
depth-map 27
21
19
20
I/O subsystem
Processing Unit 18
GPU 28

Figure 2a

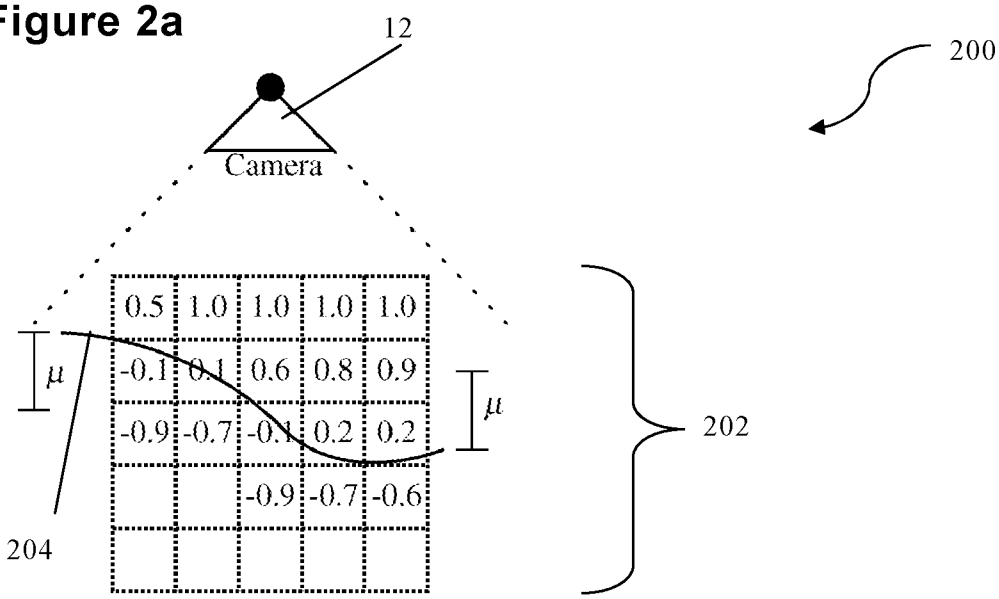


Figure 2b

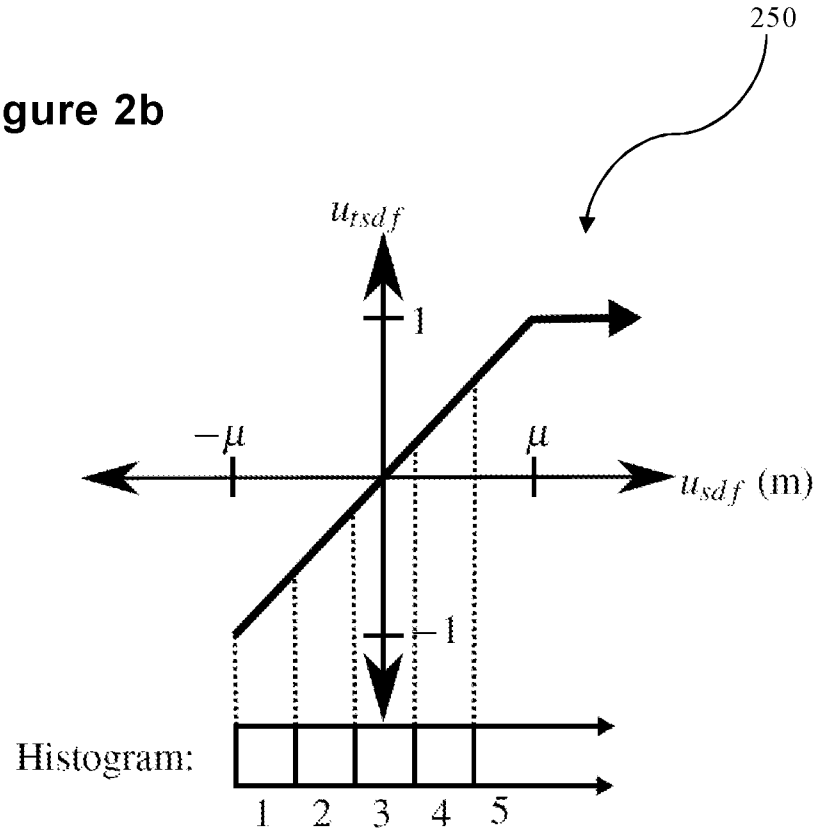


Figure 3a (Prior Art)

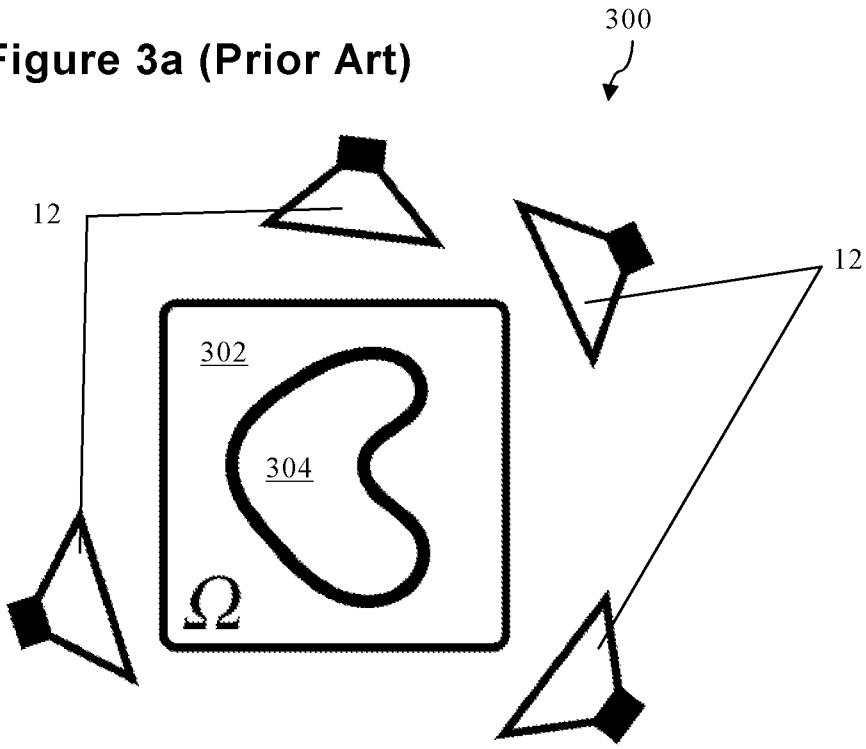


Figure 3b

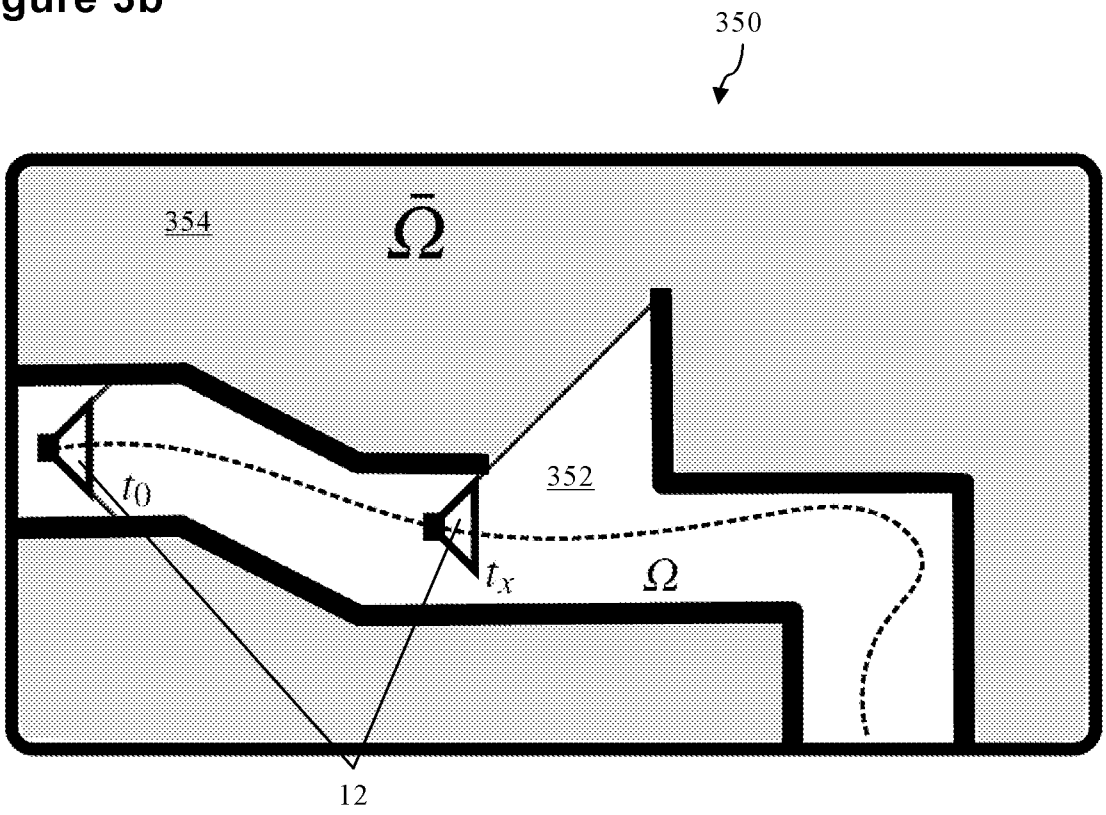
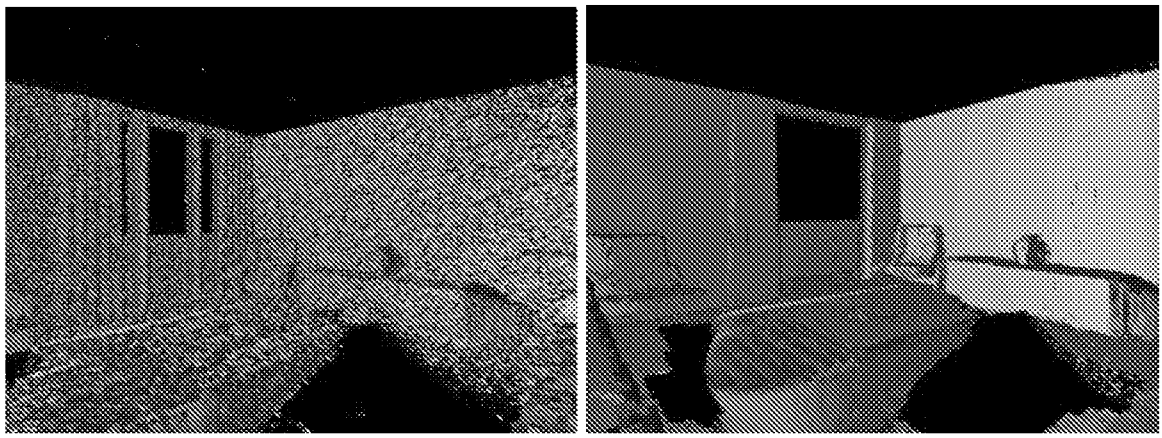


Figure 4



A B

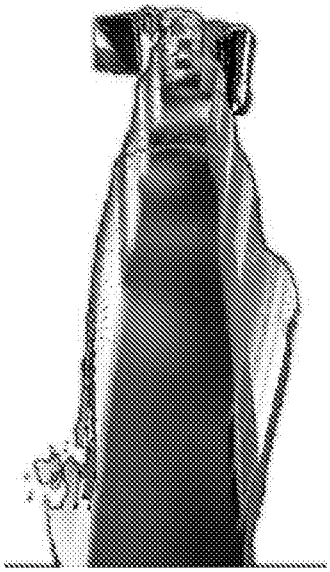


Fig. 8a

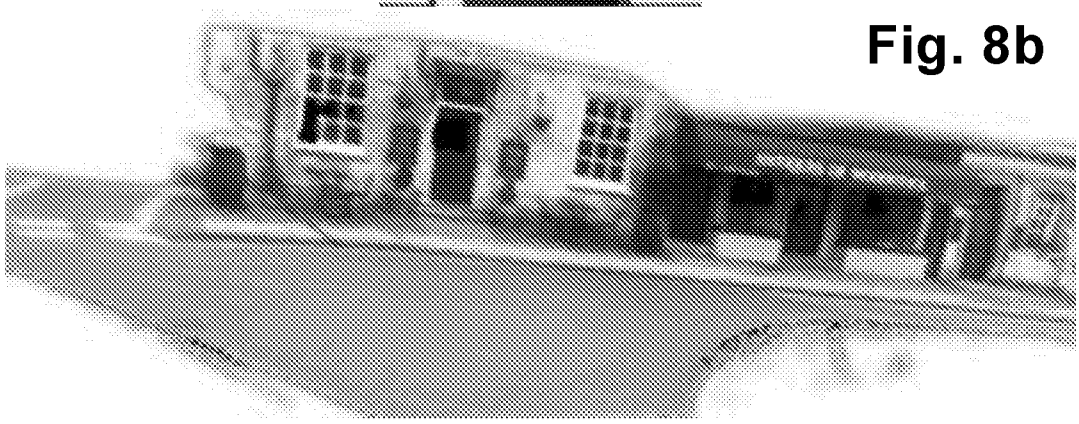


Fig. 8b

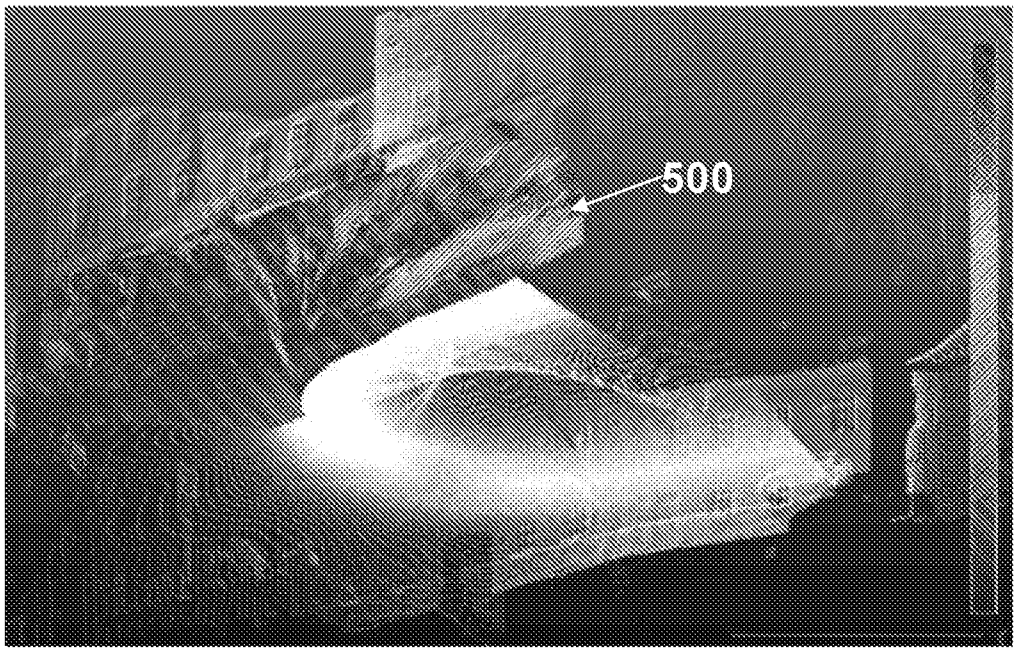


Fig. 5a

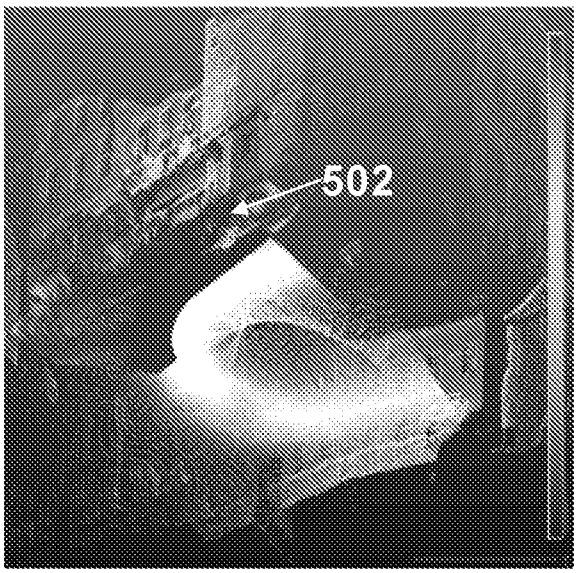


Fig. 5b

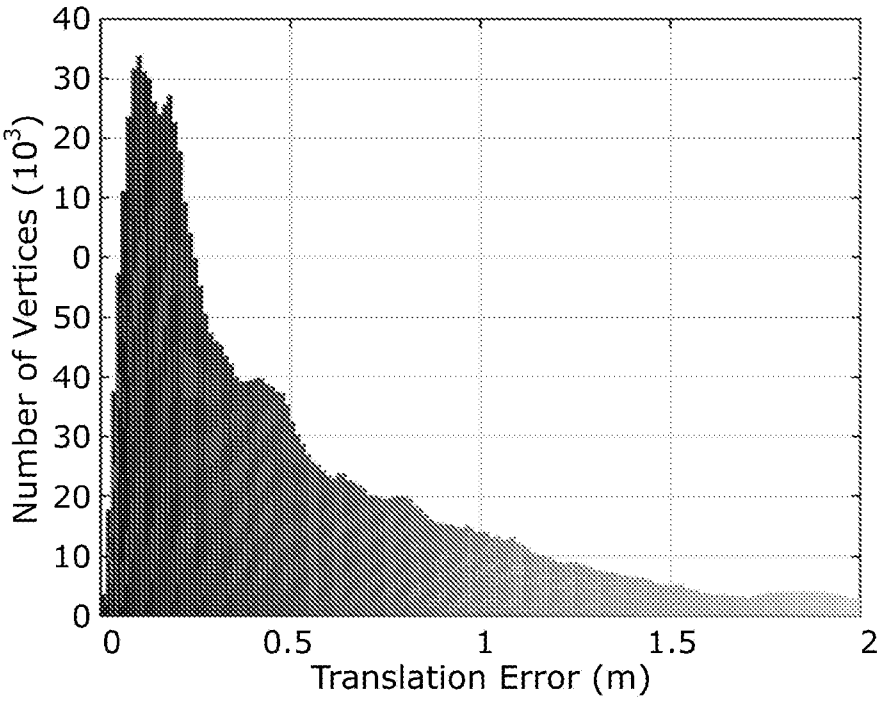

Number of Vertices (10^3)
40
30
20
10
0
50
40
30
20
10
0
0
0.5
1
1.5
2
Translation Error (m)

Fig. 5c

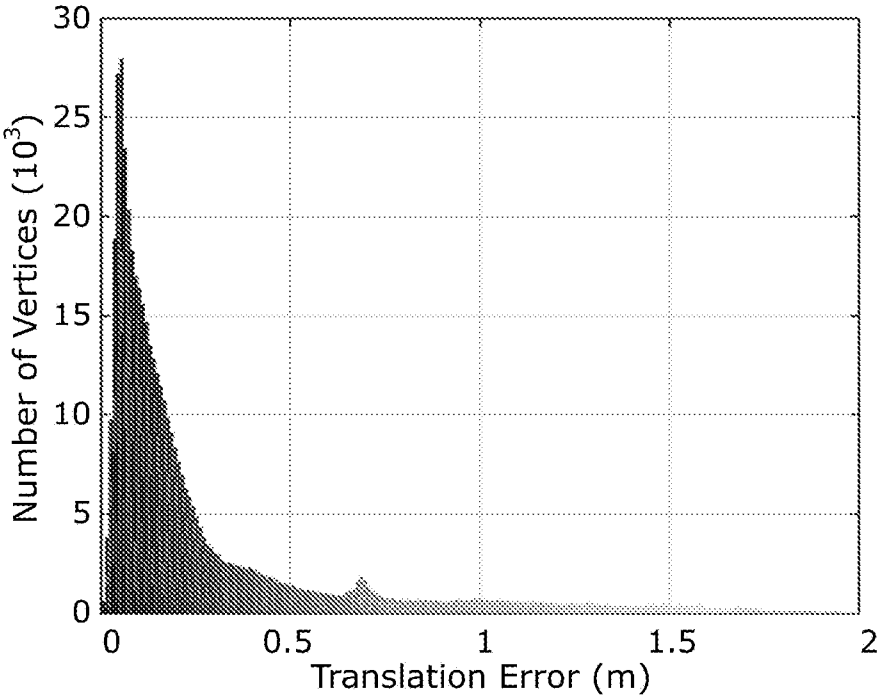

Number of Vertices (10^3)
30
25
20
15
10
5
0
0
0.5
1
1.5
2
Translation Error (m)

Fig. 5d

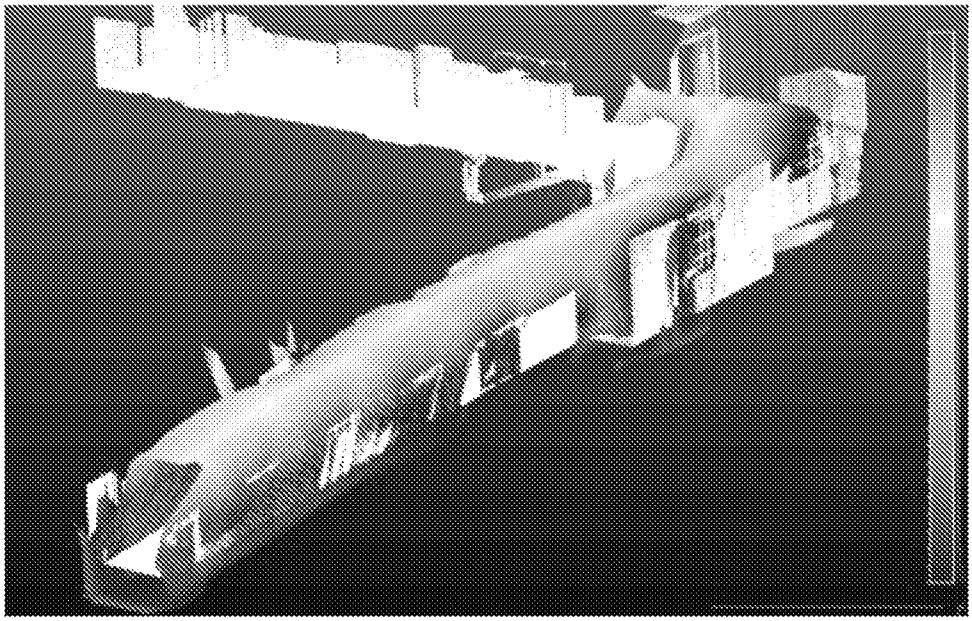


Fig. 6a

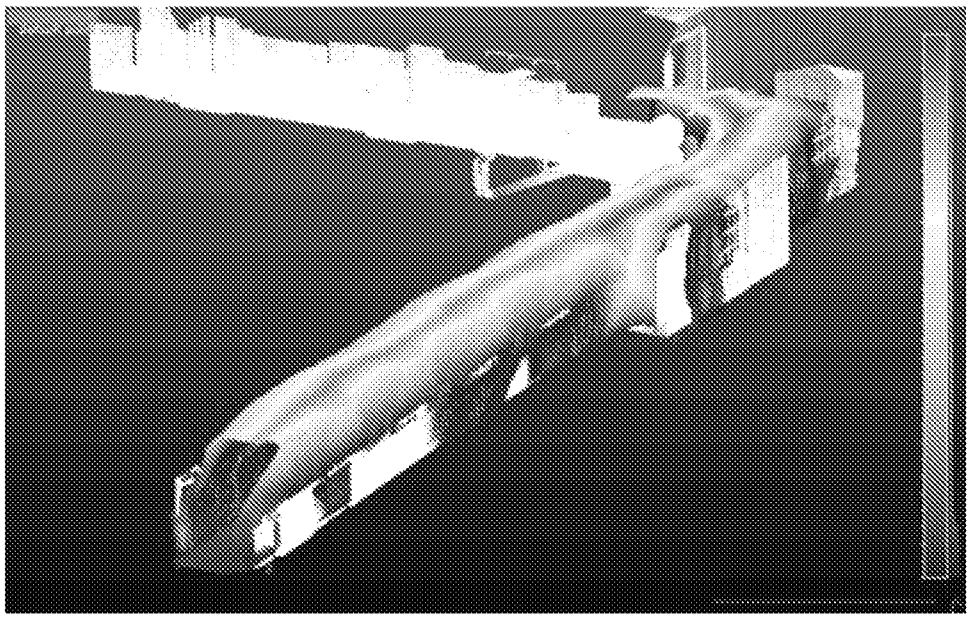


Fig. 6b

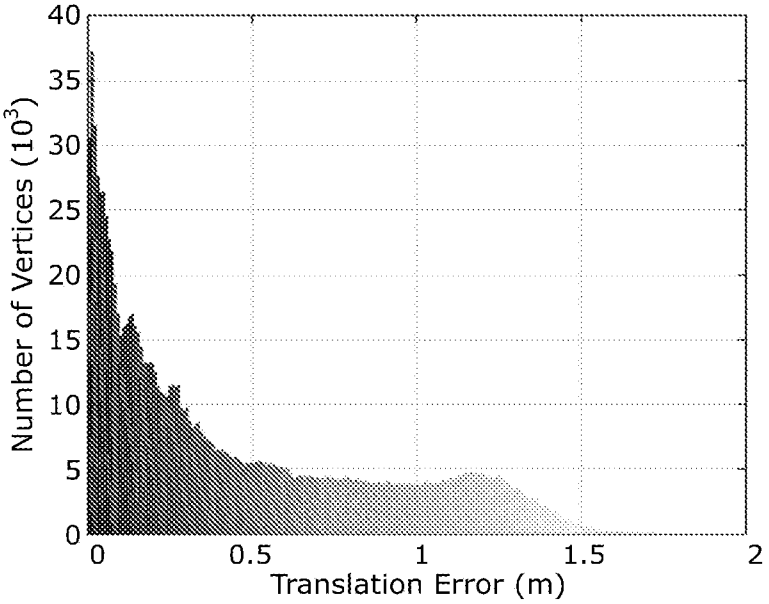

Number of Vertices (10^3)
0
5
10
15
20
25
30
35
40
0
0.5
1
1.5
2
Translation Error (m)

Fig. 6c

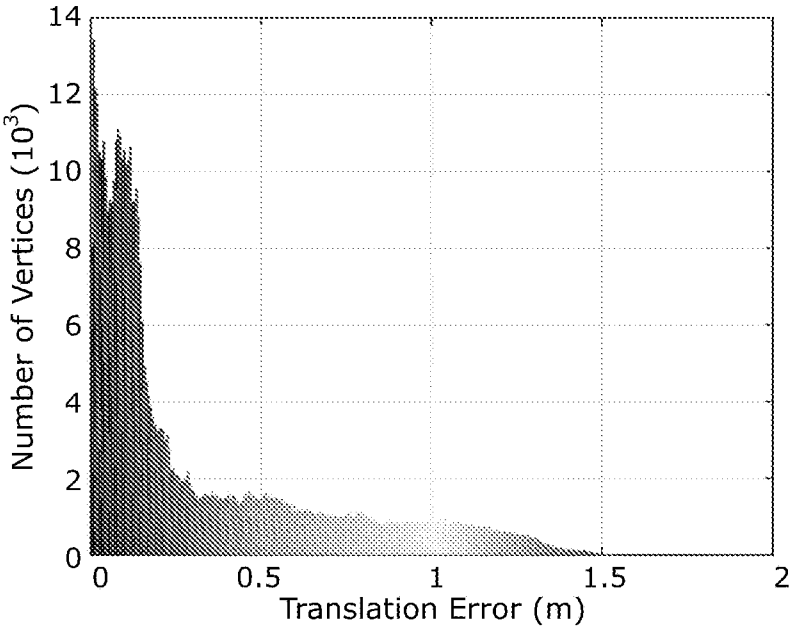
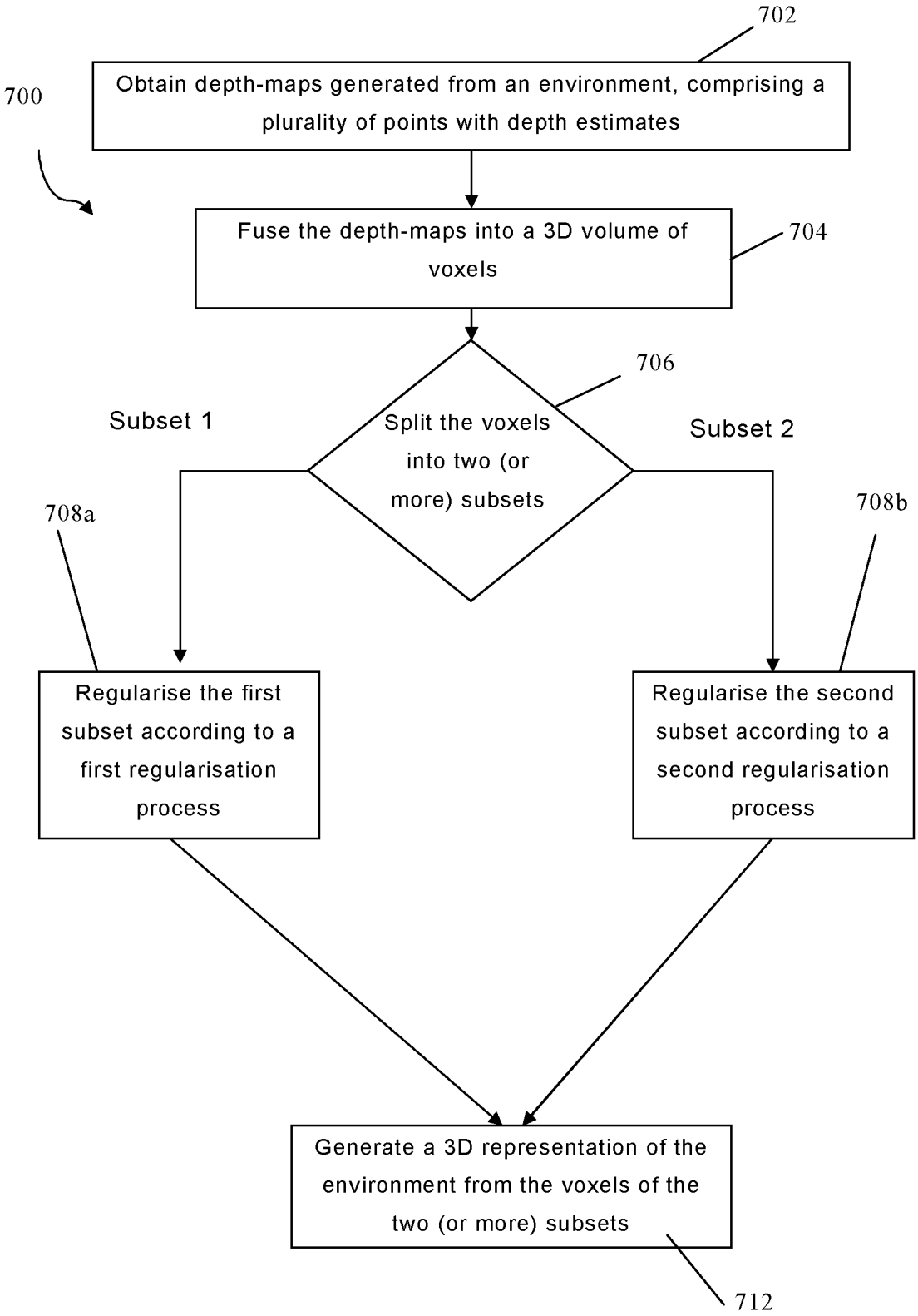

Number of Vertices (10^3)
0
2
4
6
8
10
12
14
0
0.5
1
1.5
2
Translation Error (m)

Fig. 6d

Figure 7


700
702
Obtain depth-maps generated from an environment, comprising a plurality of points with depth estimates
Fuse the depth-maps into a 3D volume of voxels
704
706
Split the voxels into two (or more) subsets
Subset 1
Subset 2
708a
708b
Regularise the first subset according to a first regularisation process
Regularise the second subset according to a second regularisation process
Generate a 3D representation of the environment from the voxels of the two (or more) subsets
712