



(51) International Patent Classification:

G06T 7/00 (2017.01) G06T 7/73 (2017.01)  
G06T 7/11 (2017.01) G06T 17/00 (2006.01)  
G06T 7/70 (2017.01)

(21) International Application Number:

PCT/GB2021/050770

(22) International Filing Date:

29 March 2021 (29.03.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2004673.6 31 March 2020 (31.03.2020) GB

(71) Applicant: **IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE** [GB/GB]; Exhibition Road, London SW7 2AZ (GB).

(72) Inventors: **WADA, Kentaro**; c/o Department of Computing, Huxley Building, 180 Queen's Gate, London SW7 2RH (GB). **SUCAR ESCAMILLA, Edgar Antonio**; c/o Department of Computing, Huxley Building, 180 Queen's Gate, London SW7 2RH (GB). **JAMES, Stephen Lloyd**; c/o Department of Computing, Huxley Building, 180 Queen's Gate, London SW7 2RH (GB). **LENTON, Daniel James**;

c/o Department of Computing, Huxley Building, 180 Queen's Gate, London SW7 2RH (GB). **DAVISON, Andrew**; c/o Department of Computing, Huxley Building, 180 Queen's Gate, London SW7 2RH (GB).

(74) Agent: **EIP**; Fairfax House, 15 Fulwood Place, London Greater London WC1V 6HU (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

(54) Title: IMAGE PROCESSING SYSTEM AND METHOD

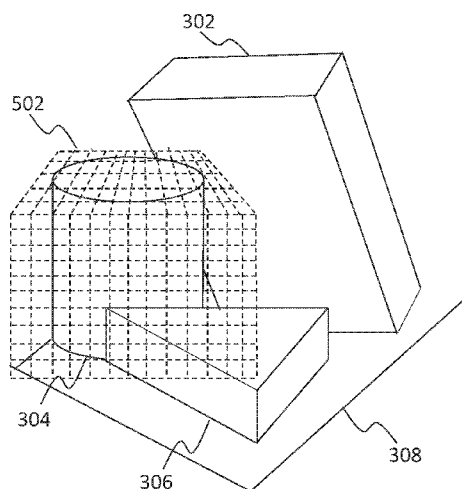


Fig. 5

(57) Abstract: A computer-implemented method of estimating a pose of a target object in a three-dimensional scene includes: obtaining image data and associated depth information representing a view of the three-dimensional scene; processing the image data and the associated depth information to generate a volumetric reconstruction for each of a plurality of objects in the three-dimensional scene, including the target object; determining a volumetric grid containing the target object; generating, using the generated volumetric reconstructions, occupancy data indicating portions of the volumetric grid occupied by free space and portions of the volumetric grid occupied by objects other than the target object; and estimating the pose of the target object using the generated occupancy data and pointwise feature data for a plurality of points on a surface of the target object.



TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,  
KM, ML, MR, NE, SN, TD, TG).

**Published:**

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

## IMAGE PROCESSING SYSTEM AND METHOD

### Technical Field

The present invention relates to processing image data and associated depth  
5 information to determine poses of objects in a three-dimensional scene.

### Background

In situations where a robot is used to manipulate or otherwise interact with  
physical objects in an environment, it is important for the robot to determine precisely  
10 the positions and orientations of the physical objects relative to a given co-ordinate  
system. This task referred to as pose prediction. Pose prediction is relevant for a  
number of other fields, for example in automated diving systems (ADS) for automated  
vehicles or advanced driver assistance systems (ADAS), where the knowing pose of an  
entity such as a vehicle or pedestrian is useful for predicting how that entity will behave.

15 Some of the earliest examples of methods for pose prediction are template-based  
methods, in which templates of an object are derived from images taken from different  
viewpoints during an offline training stage, then scanned across an image containing  
the object at test time to find a best match according to a predetermined distance metric.  
Further examples include sparse feature-based methods, in which scale-invariant points  
20 of interest are extracted from images of an object at training time and associated with  
local descriptors such as SIFT or SURF. The local descriptors are matched to an image  
containing the object at test time using a method such as RANSAC.

Recent advances in sensor technology, including for example stereoscopic  
cameras, infrared cameras, sound navigation ranging (sonar), and light detection and  
25 ranging (LIDAR) systems, allow for accurate depth information to be captured  
alongside conventional two-dimensional images, for example resulting in the RGB-D  
image format. This depth information is leveraged by certain pose prediction methods  
for improved accuracy. Examples include dense methods in which a three-dimensional  
point cloud for an object is constructed at test time and then matched to a stored model  
30 of the object using an algorithm such as Iterative Closest Point (ICP). Further examples  
include hybrid methods which simultaneously process point cloud information using a  
neural network and RGB image data using a convolutional neural network (CNN), then

fuse the outputs of the networks to derive pixel-wise dense feature embeddings which can be used for pose estimation.

In cases where multiple objects in an environment are in contact with one another, or where some of the objects are partially occluded, the accuracy of pose prediction using any of the above methods is typically reduced. In the case of robotics applications, this reduced accuracy can impair the performance of the robot, particularly in cluttered or otherwise densely populated environments.

### Summary

According to a first aspect, there is provided a computer-implemented method of estimating a pose of a target object in a three-dimensional scene. The method includes: obtaining image data and associated depth information representing a view of the three-dimensional scene; processing the image data and the associated depth information to generate a volumetric reconstruction for each of a plurality of objects in the three-dimensional scene, including the target object; determining a volumetric grid containing the target object; generating, using the generated volumetric reconstructions, occupancy data indicating portions of the volumetric grid occupied by free space and portions of the volumetric grid occupied by objects other than the target object; and estimating the pose of the target object using the generated occupancy data and pointwise feature data for a plurality of points on a surface of the target object.

By using a combination of pointwise feature data for points on the surface of the target object with occupancy data indicating surrounding regions occupied free space and other objects, the estimated pose is made dependent on detailed visual information relating to the target object itself, whilst also taking into account information relating to the surroundings of the target object. As a result, the accuracy of pose prediction is improved compared with existing methods, especially in cluttered or densely populated scenes.

In examples, the method includes processing the obtained image data and the associated depth information using a two-dimensional feature extractor to generate the pointwise feature data for the plurality of points on the surface of the target object.

In some examples in which the obtained image data and the associated depth information is processed using a two-dimensional feature extractor, the method

includes processing the obtained image data using image segmentation to determine a two-dimensional mask for the target object, and processing a portion of the image data corresponding to the two-dimensional mask for the target object using the two-dimensional feature extractor.

5           The use of masked image data means that the pointwise feature data depends only on the target object itself, without any dependence on the surroundings of the target object. In this way, the dependence of the estimated pose on the surroundings is made to depend exclusively on occupancy data for a volumetric grid surrounding the target object. In this way, information from the surroundings is taken into account in an  
10 efficient manner, without the need to process additional image data which contains information that is extraneous to the pose prediction task.

          In examples, estimating the pose of the target object using the generated occupancy data and the pointwise feature data includes: voxelating the pointwise feature data to obtain first volumetric feature data; processing the occupancy data and  
15 the first volumetric feature data together using a three-dimensional feature extractor to generate second volumetric feature data; and estimating the pose of the target object using the second volumetric feature data.

          By performing two-dimensional feature extraction and voxelating the resulting pointwise feature data, every relevant pixel of the image data contributes to the first  
20 volumetric feature data. Processing the first volumetric feature data together with the occupancy data, instead of processing the image data together with the occupancy data, results in effective use of the information-rich image data without the computational cost of performing three-dimensional feature extraction becoming prohibitive.

          In examples, three-dimensional feature extraction is performed using a three-  
25 dimensional CNN, and the second volumetric feature data comprises a hierarchy of volumetric features generated at respective different layers of the three-dimensional CNN. When the three-dimensional CNN is properly trained, the hierarchical features generated at different stages capture different latent information relevant to the estimated pose of the target object, for example with earlier stages capturing low-level  
30 features and later stages capturing high-level features.

          In some examples in which second volumetric feature data is generated, the pointwise feature data is first pointwise feature data, and estimating the pose of the

target object using the second volumetric feature data includes: extracting, from the second volumetric feature data, second pointwise feature data for the plurality of points on the surface of target object; determining, using the second pointwise feature data, a candidate pose and a corresponding confidence score for each of the plurality of points  
5 on the surface of the target object; and estimating the pose of the target object from the determined candidate poses on the basis of the corresponding confidence scores.

In examples, estimating the pose of the target object from the candidate poses comprises determining a candidate pose with a highest corresponding confidence score.

In examples, determining the candidate pose and the corresponding confidence  
10 score for each of the plurality of points on the surface of the target object includes processing the first pointwise feature data and the second pointwise feature data together using a regression model.

In examples, generating the volumetric reconstruction for a first object of the plurality of objects includes processing the image data using image segmentation to  
15 determine a two-dimensional mask for the first object, and processing a portion of the associated depth information corresponding to the two-dimensional mask to generate the volumetric reconstruction for the first object.

In examples, a method includes estimating a pose of each given target object of a plurality of target objects in a three-dimensional scene using any of the methods  
20 described above.

In some examples in which poses are estimated for a plurality of target objects, the occupancy data generated for each given target object of the plurality of target objects is first occupancy data, and the method includes, for each given target object of the plurality of target objects: sampling a plurality of points from a predetermined  
25 model of the given target object transformed in accordance with the estimated pose of the given target object; determining respective second occupancy data dependent on positions of the points sampled from the predetermined model of the given target object, relative to the volumetric grid containing the given target object; and determining respective third occupancy data dependent on positions of the points sampled from the  
30 predetermined models of the other target objects of the plurality of target objects, relative to the volumetric grid containing the given target object. The method further includes iteratively: determining an occupancy penalty depending on the respective

second occupancy data and the respective third occupancy data for each of the plurality of target objects; and iteratively modifying the estimated poses of the plurality of target objects using the determined occupancy penalty.

5 Defining the occupancy penalty using second and third occupancy data which depend on the positions of points sampled from the predetermined models of the target objects allows for incremental updating of the estimated poses of the target objects to avoid physically unrealistic predictions in which two or more of the target objects intersect with one another.

10 In examples, the view of the three-dimensional scene is a first view of the three-dimensional scene, the estimated pose of the or each target object is a first pose of the given object, and method includes obtaining further image data and further associated depth information representing a second view of the three-dimensional scene different to the first view of the three-dimensional scene. The method includes, for the or each target object: processing the further image data and the further associated depth  
15 information to estimate a second pose for the target object; transforming at least one of the first pose and the second pose of the target object to determine pose comparison data; processing the pose comparison data to determine whether a consistency condition is met; and when the consistency condition is determined to be met, generating a predetermined object model for the target object transformed consistently with the first  
20 pose and the second pose of the target object.

By capturing different views of the scene and comparing pose predictions resulting from different views, erroneous pose predictions, for example resulting from occlusion, can be identified and discarded. Once multiple pose estimates from different views are found to satisfy the consistency condition, an object model is spawned, which  
25 can be used for example by a robot interacting with the scene, or can be displayed for a human user. Furthermore, by capturing multiple views, a volumetric map of the scene can be built up iteratively using information from the different views, with the volumetric map containing fewer and fewer voxels in an “unknown” state as more views are captured. As a result, later pose predictions (which may use information from  
30 multiple views) may be more accurate than the initial pose predictions (which only use information from a single view).

According to a second aspect, there is provided an image processing system comprising a one or more sensors operable to capture image data and associated depth information. The system is arranged to: capture, using the one or more sensors, image data and associated depth information representing a view of the three-dimensional scene; process the image data and the associated depth information to generate a volumetric reconstruction for each of a plurality of objects in the three-dimensional scene, including a target object; determine a volumetric grid containing the target object; generate, using the generated volumetric reconstructions, occupancy data indicating portions of the volumetric grid occupied by free space and portions of the volumetric grid occupied by objects other than the target object; and estimate a pose of the target object using the generated occupancy data and pointwise feature data for a plurality of points on a surface of the target object.

In examples, the system includes engaging means for engaging the target object in dependence on the estimated pose of the target object. Engaging means can include one or more robotic hands or other components for grabbing, pushing, or otherwise physically contacting the target object. By engaging the target object in dependence on a pose estimated as described above, the system is able to perform intricate or sensitive tasks in a precise manner with minimal lag being introduced by the pose estimation method. In further examples, a system can interact with a target object without directly contacting the target object.

According to a third aspect, there is provided a computer program product including machine-readable instructions which, when executed by a computing system, cause the computing system to perform any of the methods described above.

Further features and advantages of the invention will become apparent from the following description of preferred embodiments of the invention, given by way of example only, which is made with reference to the accompanying drawings.

#### Brief Description of the Drawings

Figure 1 is a schematic block diagram showing a system arranged to perform methods in accordance with examples;

Figure 2 is a flow diagram representing a method of estimating a pose of a target object in accordance with examples;

Figure 3 shows an example of a three-dimensional scene comprising three known objects;

Figure 4 shows schematically a data processing routine for generating a volumetric map of a scene in accordance with examples;

5 Figure 5 shows an example of a voxel grid containing one of the known objects in the three-dimensional scene of Figure 3

Figures 6A and 6B show schematically a data processing routine for estimating a pose of a target objects in accordance with examples;

10 Figure 7 is a flow diagram representing a method of refining pose estimates for multiple objects in accordance with an example; and

Figure 8 shows an example of refinement of pose estimates for the known objects in the three-dimensional scene of Figure 3.

#### Detailed Description

15 Figure 1 shows an example of a system 100 arranged to perform methods in accordance with the present disclosure. The system 100 includes memory 102 and processing circuitry 104, where the memory 102 is arranged to store data and routines for performing the methods described herein. The processing circuitry 104 is configured to execute the routines stored in the memory circuitry 102. The processing  
20 circuitry 104 is further arranged to control sensors 106 for capturing image data and associated depth information, and one or more actuators 108 arranged to move the sensors 106.

The memory 102 in this example holds a master routine, a pose estimation routine and a pose refinement routine, along with various other routines (not shown) in  
25 the form of machine-readable instructions. In a particular configuration, execution of the master routine causes the pose estimation routine to be executed followed by the pose refinement routine, as will be described in more detail hereafter. The memory 102 further includes trainable model parameters for various trainable models used during execution of the pose estimation routine.

30 The memory 102 is arranged to hold image data and associated depth information captured by the sensors 106. In this example, the memory 102 is arranged to store image data and associated depth information in red green blue-depth (RGB-D)

format, though the system 100 can be configured for use with other suitable formats, for example based on the cyan, magenta, yellow, key (CMYK) or YUV colour formats.

The memory 102 holds a database of three-dimensional models of various known objects. In this example, the known objects are treated as rigid objects and the three-dimensional model stored for each known object is a computer aided design (CAD) model stored as a mesh representation. In the present example, a volumetric solid representation of each object is generated from the CAD model and stored alongside the CAD model. The solid representation of each object includes internal structure of the object in addition to the surface structure included for the mesh model. It is noted that for objects with complex internal structures, the internal structure of the stored solid representation does not necessarily correspond to the actual internal structure of the object, and may for example be simplified. Storing a solid model of each known object allows for points to be sampled from throughout the volume of the object, as opposed to just the surface, allowing for particularly effective implementation of pose refinement methods in accordance with aspects of the present disclosure. Nevertheless, the methods described herein can be adapted to be implemented using only mesh models, without departing from the scope of the invention.

The sensors 106 in the present example include a camera for capturing two-dimensional images of a scene and an infrared sensor for determining distances to objects in the scene (in other words, associated depth information).

Figure 2 shows an example of a computer-implemented method 200 performed by the system 100 for estimating a pose of a target object in a three-dimensional scene. The system 100 obtains, at 202, an image with associated depth information. The image represents a view of the three-dimensional scene, and the associated depth information specifies distances to different objects within the scene. In this example, the image and the associated depth information are captured using the sensors 106.

The three-dimensional scene contains multiple three-dimensional objects, at least one of which is a known object which the system 100 has been trained to recognise. In this example, the or each known object corresponds to an object model stored in the object model database in the memory 102. The scene may also include unknown objects which the system 100 has not been trained to recognise and which do not have corresponding object models stored in the memory 102. Typical examples of unknown

objects include surfaces on which the known objects are positioned, along with objects which are not relevant to the specific task which the system 100 is being used to perform.

Figure 3 shows a view of a three-dimensional scene 300. The scene 300 includes three known objects: a cuboid box 302; a cylindrical tin 304; and a prism 306 with a scalene triangular cross-section. Each known object has a three-dimensional shape and surface details including colours and textures (not shown), and in some cases including further surface decoration such as the arrow 310 on the box 302 and the symbol 312 on the tin 304. The scene 300 also includes a horizontal surface 308 on which the known objects are positioned. The horizontal surface 308 is an example of an unknown object.

The actual pose (position and orientation) of each known object in the scene 300 is represented in Figure 3 by a respective set of three-dimensional axes. The pose is a six-dimensional quantity, with three of the dimensions defining the position of a predetermined point fixed in relation to the object with respect to a given co-ordinate system, and with the other three dimensions defining the orientation relative to the given co-ordinate system. In the present example, the position is defined using Cartesian co-ordinates and the orientation is defined using extrinsic Euler angles. Those skilled in the art will appreciate that other definitions are possible, for example using cylindrical or spherical polar co-ordinates for position, and/or using intrinsic Euler angles for orientation. As will be explained in more detail hereafter, the method 200 can be used to estimate the pose of any of the known objects in the scene 300.

In the example of Figure 3, none of the known objects exhibits reflective symmetry in any plane, and therefore the pose of each known object can be defined uniquely with respect to a given co-ordinate system (note that symmetry of an object introduces ambiguity in the definition of the pose, an issue which is dealt with in certain embodiments as described hereinafter). In the present example, the tin 304 and the prism 306 are in upright orientations, each having one axis pointing vertically upwards and two axes lying in the horizontal plane. The box 302 is leaning on the tin 304 and is therefore not in an upright orientation.

Returning to Figure 2, the system 100 processes, at 204, the image and the associated depth information to generate a volumetric map. The generated volumetric

map is formed of volumetric reconstructions of the objects within the three-dimensional scene. The volumetric reconstruction of a given object includes voxels of the volumetric map corresponding to visible parts of the object in the image. From a single view of the scene, only certain parts each object are visible, and therefore only voxels  
5 corresponding to these parts will be included within the volumetric reconstruction of the object.

Figure 4 shows an example of method 400 of processing an image 402 and associated depth information 404 representing a view of the three-dimensional scene to generate a volumetric map of the scene. The image 402 is processed at 406 using image  
10 segmentation to generate an object mask 408 for each of the known objects in the scene. The object mask 408 for a given object specifies a subset of the pixels of the image 402 in which a part of the object is visible. During image segmentation, the image is partitioned into regions corresponding to the respective different object masks for the known objects in the image, along with further regions representing unknown objects  
15 within the image. In the present example, image segmentation is performed using a deep CNN architecture as described in the article Mask R-CNN by Kaiming He *et al*, arXiv: 1703.06870. Other methods for image segmentation are known and compatible with the present disclosure.

The depth information 404 and the object masks 408 are processed together at  
20 410 using volumetric fusion to generate a volumetric map 412. The volumetric map 412 includes a volumetric reconstruction 414 of each known object in the scene, and may further include volumetric reconstructions 416 of unknown objects in the scene. The depth information 404 is typically of lower resolution than the image 402, and the volumetric map 412 is typically also of lower resolution than the image 402.

Returning to Figure 2, the system 100 determines, at 206, a volumetric grid  
25 formed of a subset of the voxels of the volumetric map and containing the target object for which the pose is to be estimated. Figure 5 shows an example in which a volumetric grid 502 contains the cylindrical tin 304 in the three-dimensional scene 300. In this example, the volumetric grid 502 is a cuboid which extends beyond the tin 304 in every  
30 direction. The dimensions of the volumetric grid generally depend on the dimensions of the target object. For example, a volumetric grid containing the prism 306 would have a reduced height compared with the volumetric grid 502 containing the tin 304.

However, in order to simplify various subsequent processing steps, the number of voxels in the voxel grid for the target is fixed and independent of the dimensions of the target object (for example, 32x32x32 voxels), and the dimensions of the voxels themselves are varied to achieve different sizes and shapes of voxel grid. In order to  
5 do this, a voxel of the volumetric map can be divided into two or more smaller voxels, or two or more voxels of the volumetric map can be combined to form a single larger voxel.

The system 100 generates, at 208, occupancy data indicating portions of the volumetric grid which are occupied by free space or by objects other than the target  
10 object. Each of the voxels of the volumetric grid can be in any one of four states, depending on the occupancy of the voxel:

1. occupied by the volumetric reconstruction of the target object (e.g. the tin 304);
2. occupied by the volumetric reconstruction of one of other objects, including  
15 known objects (e.g. the box 302 and the prism 306) and unknown objects (e.g. the surface 308);
3. occupied by free space, as identified by depth measurement; or
4. unknown, because of occlusion and/or sensor range limit.

20 The voxels in states 2 and 3 are of particular interest for estimating the pose of the target object, as these voxels define an impenetrable region which cannot be occupied by any part of the target object. By indicating portions of the volumetric grid which are occupied by free space or by objects other than the target object, the occupancy data therefore includes information relevant for estimating the pose of the target object.

25 The system 100 estimates, at 210, the pose of the target object using the occupancy data generated at 208 and pointwise feature data for points on a visible portion of the target object. The pointwise feature data is derived from the pixels of image and can depend on all visual aspects of the target object, including the shape, surface details and any other information contained within the portion of the image  
30 containing the target object.

By combining the pointwise feature data for the target object with occupancy data for a voxel grid containing the target object, the estimated pose can be made

dependent on detailed visual information relating to the target object itself, whilst also taking into account information relating to the surroundings of the target object. This results in improved accuracy of pose estimation compared with known pose estimation methods.

5            Figures 6A and 6B show an example of a method 600 for estimating a pose of a target object using occupancy data and pointwise feature data. Masked image data 602 is processed at 604 using two-dimensional feature extraction to generate two-dimensional feature data 606. The masked image data 602 contains a subset of the pixels of the image 402 contained within an object mask for the target object. In this  
10            example, the two-dimensional feature extraction is performed using a CNN. In order for the masked image data 602 to be used as an input for the CNN, the masked image data should have a fixed size and shape. In order to achieved this, pixel values for pixels outside of the object mask are set to a constant value (for example, zero). The use of masked image data means that the two-dimensional feature data 606 depends  
15            only on the target object itself, without any dependence on the surroundings of the target object. In this way, the dependence of the estimated pose on the surroundings is made to depend exclusively on occupancy data for a volumetric grid surrounding the target object, as will be explained in more detail hereafter.

              The two-dimensional feature data 606 is processed, along with masked depth  
20            information 608, using pointwise encoding at 610, to generate pointwise feature data 612. The pointwise feature data 612 includes multiple feature channels for each of a set of three-dimensional points derived from the masked depth information 608. The points form a point cloud representing portions of the object visible in the image. In the present example, the two-dimensional feature data 606 and the masked depth  
25            information 608 are processed separately using respective fully connected neural network layers, and the resulting pointwise features are concatenated to generate the pointwise feature data 612.

              The pointwise feature data 612 is processed at 614 using voxelisation, to  
30            generate a feature grid 616. The voxelisation (also known as voxelation) associates points specified in the pointwise feature data 612 with voxels of the voxel grid containing the target object (for example, the voxel grid 502 in Figure 5). The voxel grid typically has a relatively low resolution (for example, 32x32x32 voxels) compared

with the image and the associated depth data. The feature data for a given voxel is therefore derived from multiple points of the pointwise feature data 612, in this example by averaging, though in other examples the feature data for a given voxel is derived using other methods, for example by taking maximum values. Each voxel of the feature  
5 grid has an associated volumetric index specifying its location in the voxel grid, and these indices 618 are stored for subsequent processing.

The feature grid 616 is concatenated with occupancy data 620 indicating regions of the voxel grid which cannot be occupied by the target object because they are occupied by other objects or free space. The occupancy data 620 associates a binary  
10 number to each voxel of the voxel grid containing the target object, where the binary number indicates whether that voxel is impenetrable to the target object (i.e. whether the voxel is in either of states 2 or 3 referred to above). The concatenated feature grid 616 and occupancy data 620 therefore include, for each voxel of the voxel grid, a binary channel from the occupancy data 620 and multiple channels from the feature grid 616.  
15 The concatenated feature grid 616 and occupancy data 620 therefore contain information derived from the masked image data 602 and masked point cloud 610, and further contains information depending on the objects and space surrounding the target object.

The concatenated feature grid 616 and occupancy grid 620 are processed at 622  
20 using three-dimensional feature extraction to generate three-dimensional feature data 624. In this example, the three-dimensional feature extraction is performed using a three-dimensional CNN having multiple stages each containing several convolutional layers. Each stage of the three-dimensional CNN generates a volumetric feature map, and after each stage a pooling or compression operation is performed to reduce the  
25 dimensionality of the volumetric feature map before processing by the next stage of the three-dimensional CNN. As a result, the three-dimensional CNN generates a hierarchy of volumetric feature maps at sequentially decreasing resolution. When the three-dimensional CNN is properly trained (as explained hereafter), the hierarchical features generated at different stages capture different latent information relevant to the  
30 estimated pose of the target object. The three-dimensional feature data 624 includes the volumetric feature maps generated at the different stages of the three-dimensional CNN. In the present example, the concatenated feature grid and occupancy grid

contains 32x32x32 voxels, the three-dimensional CNN includes three stages, and the hierarchical volumetric feature maps contain 32x32x32, 16x16x16 and 8x8x8 voxels of features respectively.

5 It is noted that, in the present example, two-dimensional feature extraction from the masked image data 602 is performed independently of the three-dimensional feature extraction at 622. By performing two-dimensional feature extraction first, every pixel of the masked image data 602 contributes to the pointwise feature data 612, resulting in effective use of the information-rich masked image data without the computational cost becoming prohibitive. In other examples, image data is processed directly  
10 alongside occupancy data using a three-dimensional feature extractor. However, this approach usually requires a reduction in resolution of the image data in order to keep the required computational resources (processing power and memory) to a reasonable level. Therefore, information contained within the image data is lost.

Points extraction is performed at 626 to extract pointwise feature data 628 from  
15 the three-dimensional feature data 624, for points corresponding to the indices 618 stored during the voxelisation at 614. For each point corresponding to one of the indices 618, the corresponding features within the three-dimensional feature data 624 are extracted and stored. The pointwise feature data 628 therefore includes features for the same set of points as the pointwise feature data 612 derived from the masked image  
20 data 602 and the masked depth information 608. The pointwise feature data 612 and the pointwise feature data 628 are concatenated for the purpose of pointwise pose estimation.

It is noted that the pointwise feature data 612 depends strongly on the visual appearance and depth profile of the target object. The pointwise feature data 628 also  
25 has some dependence on the appearance and depth profile of the target object, but further depends on the surrounding space and objects. The inventor has found that using the pointwise feature data 612 strongly dependent on the visual appearance and depth profile of the target object, in combination with the surrounding occupancy data, results in a significant improvement of the accuracy of pose detection over known  
30 methods.

Pointwise pose estimation is performed at 630 using the concatenated pointwise feature data 612 and 628. In the present example, the pointwise pose estimation

determines a candidate pose 632 and a candidate confidence score 634 for each of the points within the pointwise feature data 612 and 628. Each candidate pose 632 is a six-dimensional vector and the candidate confidence score 634 is a numerical value indicative of certainty that the corresponding candidate pose is correct. In this example, 5 the pointwise pose estimation is performed using a fully connected neural network.

An estimated pose 638 is determined at 636 as a best of the candidate poses 632 on the basis of the confidence scores 634. In other words, the estimated pose 638 is determined as the candidate pose 632 having the highest confidence score 634.

Although in the example described above, pointwise pose estimation is used to 10 determine a respective candidate pose for each of a set of points, in other examples pointwise feature data is processed to generate a single, global pose estimate, in which case there is no need for confidence scores to be determined.

The method 600 of Figures 6A and 6B makes use of a series of trainable model components, namely the two-dimensional feature extractor used at 604, pointwise 15 encoders used at 610, a three-dimensional feature extractor used at 622, and a pointwise pose estimator used at 630. In the present example, each of these components is a neural network model having a respective set of trainable parameters. Prior to the method 600 being performed, the models are trained using supervised learning with a labelled dataset of known objects with known ground truth poses. In the present 20 example, the trainable models used in the pose estimation method 600 are trained independently of the image segmentation model used to generate the masked image data 602. In view of this, for the purpose of training the pose prediction models, labelled training data is generated by positioning and orienting volumetric models of known objects (in this example, CAD models of the known objects) to generate an artificial 25 scene, for example using a physics model with collision detection. In this way, a large number of artificial scenes can be generated with perfectly segmented objects and known ground truth poses, without the time-consuming tasks of moving physical objects into position for the training of the model and manually determining the ground truth poses of the objects. Furthermore, the present approach using volumetric models 30 allows the pose estimation models to be trained rapidly and in isolation from certain expensive physical components of the system 100, including the sensors 106 and the actuators 108. In order to train the pose estimation models, a set  $X$  of points is sampled

(for example, uniformly) from a volumetric model of the target object. The positions of the points in  $X$  when transformed by the ground truth pose of the target object in the scene are compared with the positions of the points in  $X$  when transformed using each pointwise candidate pose estimate.

5           The models used in the pose estimation method 600 are trained using a single pose estimation loss  $L$ . At each of a set of training iterations, a gradient  $\nabla_{\theta}L$  of the pose prediction loss is determined with respect to the trainable parameters  $\theta$  of the pose prediction models using backpropagation, and the values of the trainable parameters  $\theta$  are updated using gradient descent or a variant thereof to reduce the value of the pose  
10 estimation loss  $L$ . This updating is performed iteratively until predetermined stopping conditions are satisfied, which may correspond to predetermined convergence criteria being satisfied or a predetermined number of training iterations being performed.

In the present example, the pose estimation loss  $L$  is given by Equation (1):

$$L = \frac{1}{N} \sum_{i=1}^N (L_i c_i - \lambda \log(c_i)), \quad (1)$$

15

where:

- $N$  is the number of points for which candidate poses are estimated;
- $c_i$  is the confidence score associated with the candidate pose of the  $i^{\text{th}}$  point;
- $\lambda$  is a scaling factor for a regularisation term  $\log(c_i)$ ; and
- 20 •  $L_i$  is a pointwise pose estimation loss associated with the candidate pose determined for the  $i^{\text{th}}$  point.

Appropriate values for  $\lambda$  have been found to be in the range  $\lambda \in [0.01, 0.1]$ , and in particular in the range  $\lambda \in [0.01, 0.02]$ , for example  $\lambda = 0.015$ . The scaling factor can be tuned manually for a given training instance or can be included as a parameter  
25 to be learned during training. The pointwise pose estimation loss in this example is given by Equation (2):

$$L_i = \frac{1}{|X|} \sum_{q=1}^{|X|} \|(Rp_q + \mathbf{t}) - (\hat{R}_i p_q + \hat{\mathbf{t}}_i)\|, \quad (2)$$

where:

- $[R|\mathbf{t}] \in SE(3)$  is the ground truth pose of the target object, consisting of a 3x3 rotation matrix  $R \in SO(3)$  and a translation vector  $\mathbf{t} \in \mathbb{R}^3$ ;
- 5 •  $[\hat{R}_i|\hat{\mathbf{t}}_i] \in SE(3)$  denotes the candidate pose estimate for the  $i^{\text{th}}$  point; and
- $p_q$  is the position of the  $q^{\text{th}}$  point of the set  $X$  of points sampled from the volumetric model of the target object.

The pointwise pose estimation loss of Equation (2) is appropriate for objects which do not exhibit reflective symmetry in any plane. For symmetric objects, an ambiguity arises as to which point transformed by the ground truth pose should be compared with a point transformed by a given candidate pose estimate. For such objects, a modified pointwise pose estimation loss is used, as given by Equation (3):

$$L_i = \frac{1}{|X|} \sum_{q=1}^{|X|} \min_{p_{q'} \in X} \|(Rp_q + \mathbf{t}) - (\hat{R}_i p_{q'} + \hat{\mathbf{t}}_i)\|, \quad (3)$$

15 which effectively results in the nearest point after transformation by the candidate pose estimate being compared with a given point transformed by the ground truth pose. In a specific configuration, a first training stage is performed using the unmodified pointwise training loss of Equation (2), followed by a second training stage using the modified pointwise training loss of Equation (3). This has been found by the inventor to avoid local minima which can sometimes result from use of the modified pointwise pose loss, whilst avoiding erroneous pose estimations which would result from the use of the unmodified pointwise training loss for symmetric objects. This results in particularly good performance of the pose estimation method for symmetric objects with complicated shapes.

25 The method 200 of Figure 2 can be used to estimate poses of multiple objects in a given scene, with the estimated pose for each given object taking into account the

occupancy of a region of space surrounding the given object. In some examples, the estimated poses are sufficiently accurate to be used for a given purpose. In other examples, it cannot be guaranteed that pose estimates determined using the method 200, or any other pose estimation routine, are accurate enough for a given purpose. In such cases, there is a need for a method of refining pose estimates, at the expense of some additional computing cost and time.

Figure 7 shows an example of a computer-implemented method 700 performed by the system 100 for estimating a pose of each of a plurality of known objects in a three-dimensional scene. For each given object of the plurality of known objects, the system 100 obtains, at 702, image data and associated depth information representing a view of a three-dimensional scene in which at least a part of the given object is visible (the given object may be partially occluded or extending beyond the boundary of the image), and estimates, at 704, a pose for the given object. In the present example, the image and associated depth information is captured using the sensors 106 and the pose of each object is estimated using the method 200 of Figure 2. Alternatively, any other suitable pose prediction methods could be used. In some examples, poses of the several objects in the scene are estimated using a single image and associated depth information. In other examples, poses of two or more different objects are estimated using different images representing different views of the three-dimensional scene. In the latter case, the estimated pose of each given object is transformed to a common reference frame (for example, using information indicative of the position and orientation of the sensors 106 when capturing the image and associated depth information).

Having determined an estimate for the pose of each of the plurality of objects, and transformed the estimated poses to a common reference frame if necessary, the system 100 performs an iterative pose refinement routine to jointly optimise the estimated poses as described hereafter. The pose refinement routine starts at 706, where the system 100 samples a set of points from a stored model of each given object, transformed in accordance with the corresponding estimated pose of the object. In this example, the stored model is a volumetric solid model and the system 100 samples the set of points uniformly from throughout the volume of the volumetric solid model. The sampled set of points for each given object forms a point cloud.

The system 100 determines, at 708, respective first occupancy data for each given object dependent on positions of the points sampled from the stored model of the given object, relative to a voxel grid containing the given object. In the present example, where the system 100 uses the method 200 to determine the initial pose estimates, the system 100 has already determined a voxel grid containing each given object. The same voxel grid is therefore reused for defining the first occupancy data. In other examples, such as when the method 200 is not used to determine the initial pose estimates, a new voxel grid is determined for each given object at 708.

The first occupancy data for a given object depends on points with locations depending on the estimated pose of the given object, relative to a voxel grid containing the given object. The positions of the points are therefore differentiable with respect to the pose of the given object. In other words, a small change in the estimated pose of the given object leads to a predictable small change in the position of each sampled point. Provided that the first occupancy data depends on the positions of the points in a smooth, differentiable manner, the first occupancy data is therefore also differentiable with respect to the estimated pose of the given object.

The system 100 determines, at 710, respective second occupancy data for each given object dependent on positions of the points sampled from the stored models of the other objects of the plurality of objects, relative to the voxel grid containing the given object. The second occupancy data for a given object is differentiable with respect to the estimated poses of the other objects in the same way that the first occupancy data is differentiable with respect to the estimated pose of the given object.

The system 100 updates, at 712, the estimated poses of the plurality of objects to reduce an occupancy penalty depending on the respective first occupancy data and the respective second occupancy data for each of the plurality of objects. The occupancy penalty depends on the first occupancy data and the second occupancy data in a differentiable manner, and in turn is therefore differentiable with respect to the estimated poses of the plurality of objects. This allows a gradient of the occupancy penalty to be determined with respect to the estimated poses of the plurality of objects, which in turns allows for incremental updating of the estimated poses using gradient descent of a variant thereof.

The steps 706-712 are performed iteratively until a stopping condition is satisfied. The stopping condition may include predetermined convergence criteria being satisfied, or may include a predetermined number of iterations having been performed.

- 5            In a specific example, the first occupancy data includes a first differentiable occupancy grid for each given object of the plurality of objects. The first differentiable occupancy grid  $g_m^{\text{given}}$  for the  $m^{\text{th}}$  object consists of a first differentiable occupancy value for each voxel of the voxel grid containing the  $m^{\text{th}}$  object. The first differentiable occupancy value  $o_k^m$  for the  $k^{\text{th}}$  voxel depends on a minimum distance between the  $k^{\text{th}}$  voxel and the points sampled from the volumetric model of the given object, as shown  
10 by Equation (4):

$$o_k^m = 1 - \min\left(\delta^t, \min_q(\delta_{qk}^{mm})\right), \quad (4)$$

- where  $\delta_{qk}^{mm}$  is the distance between the  $k^{\text{th}}$  voxel of the voxel grid containing the  $m^{\text{th}}$  and the  $q^{\text{th}}$  point sampled from the volumetric model of the  $m^{\text{th}}$  object, and  $\delta^t$  is a  
15 predetermined distance threshold. In the present example, the dependence on the minimum distance  $\min \delta_{qk}^{mm}$  saturates at the distance threshold, so that if no point is closer to the voxel than the distance threshold, the occupancy for that voxel is set to 0 and that voxel does not contribute to the occupancy penalty. Once any point is brought  
20 closer to the voxel than the predetermined distance threshold, the differentiable occupancy increases continuously, reaching a maximum value of 1 if the point coincides with the voxel. In order to calculate the distances  $\delta_{qk}^{mm}$ , the position  $p_q^m$  of the  $q^{\text{th}}$  point sampled from the model of the  $m^{\text{th}}$  object is transformed to the co-ordinate system of the voxel grid using the equation  $u_q^m = (p_q^m - l)/s$ , where  $l$  is an origin of the voxel grid coordinate system and  $s$  is the size of each voxel in the voxel grid. The  
25 distances are then given by  $\delta_{qk}^{mm} = |u_q^m - v_k^m|$ , where  $v_k^m$  is a position associated with the  $k^{\text{th}}$  voxel (for example, a predetermined corner of the voxel or the centre of the voxel), and  $u_q^m$  is the position of the point in the voxel coordinate system.

In this example, the second occupancy data includes a second differentiable occupancy grid for each given object of the plurality of given objects. The second occupancy grid  $g_m^{\text{other}}$  for the  $m^{\text{th}}$  object consists of a second differentiable occupancy value for each voxel of the voxel grid containing the  $m^{\text{th}}$  object. The second differentiable occupancy value  $\tilde{o}_k^m$  for the  $k^{\text{th}}$  voxel depends on a minimum distance between the  $k^{\text{th}}$  voxel and the points sampled from the volumetric models of all of the other given objects, as shown by Equation (5):

$$\tilde{o}_k^m = 1 - \min\left(\delta^t, \min_{q,n}(\delta_{qk}^{mn})\right), \quad (5)$$

where  $\delta_{qk}^{mn}$  is the distance between the  $k^{\text{th}}$  voxel of the voxel grid containing the  $m^{\text{th}}$  object and the  $q^{\text{th}}$  point sampled from the volumetric model of the  $n^{\text{th}}$  object (where  $n \neq m$ ). In order to determine the distances  $\delta_{qk}^{mn} = |u_q^n - v_k^m|$ , the points sampled from the models of the other objects are transformed to the co-ordinate system of the voxel grid containing the  $m^{\text{th}}$  object.

In the present example, the occupancy penalty  $L^o$  includes, for each given object of the plurality of known objects, a collision component  $L_m^{o+}$  which increases when a point sampled from the predetermined model of the given object and a point sampled from the predetermined model of a different object of the plurality of known objects are simultaneously brought closer to a voxel of the voxel grid containing the given object. The collision component  $L_m^{o+}$  in this example is derived from the first differentiable occupancy grid  $g_m^{\text{given}}$  and the second differentiable occupancy grid  $g_m^{\text{other}}$ , as shown by Equation (6):

$$L_m^{o+} = \frac{(g_m^{\text{given}} \circ g_m^{\text{other}})}{\sum_k o_k^m}, \quad (6)$$

where  $\circ$  denotes the elementwise product. The collision component  $L_m^{o+}$  penalises situations where a voxel of the voxel grid containing the  $m^{\text{th}}$  object is simultaneously close to a point sampled from the  $m^{\text{th}}$  object and a point sampled from one of the other

objects of the plurality of known objects. A possible definition of the overall occupancy penalty is then given by  $L^o = \sum_m L_m^{o+} / N$ , where the sum is over the  $N$  known objects. The overall occupancy penalty is optimised jointly with respect to the estimated poses of all of the known objects. More sophisticated definitions of the occupancy penalty are possible, however, as will be explained hereafter.

In some examples, such as those in which the method 200 of Figure 2 is used to determine the initial pose estimates, volumetric reconstructions are generated for each given object of the plurality of known objects, additional occupancy data is generated from volumetric fusion. This additional occupancy data indicates portions of the voxel grid containing the given object which are occupied by free space and portions of the voxel grid containing the given object which are occupied by objects other than the given object. Unlike the first occupancy data and the second occupancy data, this additional occupancy data is not differentiable with respect to the estimated poses of the objects. The additional occupancy data does, however, contain information which is relevant for pose refinement, because the additional occupancy data defines an impenetrable region of the voxel grid containing the given object. Given this information, the collision component for a given object can be modified to increase when a point sampled from the predetermined model of the given object is brought closer to a voxel of the voxel grid containing the given object which is occupied by free space or by objects other than the given object.

In one example, the additional occupancy data includes a binary impenetrable grid  $g_m^{\text{inpen}}$  which associates a binary number to each voxel of the voxel grid containing the given object, where the binary number indicates whether that voxel is impenetrable to the given object (i.e. whether the voxel is in either of states 2 or 3 referred to above). For compatibility with the definitions of the first differentiable occupancy grid  $g_m^{\text{given}}$  and the second differentiable occupancy grid  $g_m^{\text{other}}$ , the impenetrable grid  $g_m^{\text{inpen}}$  is given a value of 1 for impenetrable voxels, and 0 otherwise. It will be appreciated that other definitions are possible, however.

Given the impenetrable grid  $g_m^{\text{inpen}}$ , an alternative definition of the collision component for the  $m^{\text{th}}$  given object is given by Equation (7):

$$L_m^{o+} = \frac{(g_m^{\text{given}} \circ \max(g_m^{\text{other}}, g_m^{\text{impen}}))}{\sum_k o_k^m}, \quad (7)$$

where the maximum operator is taken elementwise. This alternative definition penalises situations where a voxel of the voxel grid containing the  $m^{\text{th}}$  object is close to a point sampled from the  $m^{\text{th}}$  object and is simultaneously close to a point which is sampled from one of the other known objects, and/or which is part of the impenetrable grid. The alternative definition of the collision component can result in improved performance of the pose refinement method, because the resulting set of poses is constrained by impenetrable regions of the volumetric map of the scene.

In addition to a collision component, defined for example by Equation (6) or Equation (7), the occupancy penalty can be augmented to include a surface alignment component for each of the plurality of known objects. Unlike the collision component, which penalises overlapping of neighbouring objects, the surface alignment component rewards situations where points sampled from a given object overlap with voxels of the volumetric reconstruction for that object. The surface alignment component therefore encourages consistency between the estimated pose of the given object and the appearance of the given object in the image and associated depth information.

In an example, the surface alignment component for the  $m^{\text{th}}$  given object is given by Equation (8):

$$L_m^{o-} = -\frac{(g_m^{\text{given}} \circ g_m^{\text{self}})}{\sum_k o_k^{m,\text{self}}}, \quad (8)$$

20

where  $g_m^{\text{self}}$  is a binary self-occupancy grid with elements given by  $o_k^{m,\text{self}}$ , where in this example  $o_k^{m,\text{self}}$  has a value of 1 for voxels occupied by the volumetric reconstruction of the  $m^{\text{th}}$  object, and 0 otherwise. The surface alignment component for a given object decreases when a point sampled from the predetermined model of the given object is brought closer to a voxel of the voxel grid containing the given object which is occupied by the volumetric reconstruction for the given object.

25

When a surface alignment component is included, the occupancy penalty is defined by Equation (9):

$$L_m^o = \frac{1}{N} \sum_{m=1}^N (L_m^{o+} + L_m^{o-}). \quad (9)$$

5           The occupancy penalty is optimised jointly with respect to the estimated poses of all of the known objects. In one example, the optimisation is performed using batch gradient descent on a graphics processing unit (GPU).

Figure 8 illustrates the effect of including a surface alignment component in the occupancy penalty. The top frame shows an initial set of estimated poses for the known  
 10           objects 302, 304, 306 in the scene 300 of Figure 3, in this example determined using the method 200 of Figure 2. It is observed that the estimated poses are inaccurate, resulting in the box 302 intersecting with the tin 304. As a result, the collision component for the tin 304 (defined using the voxel grid 502) has a high value. The lower frame on the left shows the estimated poses of the objects 302, 304, 306 after  
 15           pose refinement has been performed without the use of a surface alignment component. It is observed that the tin 304 has moved to the left such that there is no longer an overlap with the box 302, in order to reduce the collision component for the tin 304. However, the resulting scene does not correspond closely with the actual scene 300 shown in Figure 3. The lower frame on the right shows the estimated poses of the  
 20           objects 302, 305, 306 after pose refinement has been performed using a surface alignment component. It is observed that the angle of the box 302 has been modified such that there is no longer an overlap with the tin 304, in order to reduce the collision component of the occupancy penalty. However, in contrast with the erroneous scene in the lower left frame, the resulting scene in the lower right frame corresponds closely  
 25           with the scene 300 in Figure 3, because the surface alignment component for the tin 304 has prevented the tin 304 from moving away from its original position in a way which is inconsistent with its volumetric reconstruction. The surface alignment component thereby results in more accurate pose prediction.

Using the methods 200 and/or 700, the system 100 is able to predict a first pose for an object in a scene using an image and associated depth information representing a first view of the scene captured by the sensors 106. However, the entirety of the object will not be visible from any single view of the scene. In order to achieve even more accurate pose prediction, in the present example the system 100 is further configured to move the sensors 106 using the actuators 108 to capture a further image and associated depth information representing a second view of the scene. The second view is different from the first view because the sensors 106 have a different orientation and/or position relative to the scene after being moved.

Using the further image and associated depth information, the system 100 predicts a second pose for the object. In the present example, the second pose is predicted using the same method as the first pose, namely the pose estimation method 200 followed by the pose refinement method 700. At least one of the first pose and the second pose is transformed such that the first pose and the second pose are expressed with respect to a common coordinate system. In the present example, both the first pose and the second pose are transformed to an arbitrary “world” coordinate system which is independent of the position and orientation of the sensors 106.

Transforming the first pose and/or the second pose to a common coordinate system allows the first pose and the second pose to be compared. If, on the basis of this comparison, a consistency condition is determined to be met, the first pose and second pose are determined to be accurate. If the consistency condition is not determined to be met, a further image and associated depth information is captured representing a third view of the scene, which is then compared with each of the first pose and the second pose. If the third pose is consistent with either the first pose or the second pose, then that pose is determined to be accurate. Further images and associated depth information are captured, and further poses predicted for the object, until the consistency condition is satisfied. In the present example, the poses are compared using the pointwise pose estimation loss of Equation (2), and the consistency condition is satisfied when any two predicted poses with a pointwise pose estimation loss of less than a threshold value. In other examples, the consistency condition is satisfied when a threshold number  $M$  of predicted poses have a pointwise pose estimation loss  $L_i$  of less than a threshold value  $L^t$ , i.e. when  $M = \text{count}(L_i < L^t)$ .

If a pose prediction is determined to be accurate on the basis of pose comparisons as described above, the system 100 spawns a mesh model of the object transformed consistently with the predicted pose. By spawning mesh models of multiple known objects, a mesh model of the scene is generated. The mesh model of the scene can be used for interacting with the scene (as described in more detail hereafter), or can be displayed for a human user.

It is noted that, during the process of capturing different views of the scene and making further pose predictions, the volumetric map of the scene can be built up iteratively using information from the different views, with the volumetric map containing fewer and fewer voxels in the “unknown” state as more views are captured. As a result, later pose predictions (which use information from multiple views) are likely to be more accurate than the initial pose predictions (which only use information from a single view). However, capturing multiple views of the scene takes additional time. Therefore, a trade-off arises between accuracy and time. In some cases, it is essential that pose prediction is performed quickly, in which case it may be necessary to predict the poses of a given object from only a single view of a scene. Examples include pose prediction performed by an ADS or ADAS in a vehicle. In other cases, accuracy is of paramount importance. Examples of such cases include very intricate robotics tasks.

The methods described herein are particularly valuable for robotics tasks in which a robot is used to pick up or otherwise engage objects. Such a robot includes one or more engaging means such as robotic hands or other components for grabbing, pushing, or otherwise physically contacting a given object. In order to correctly engage the given object, the robot first predicts the pose of the given object and then engages the given object in dependence on the predicted pose. In some examples, the robot first spawns a mesh model of the given object (for example, a CAD model) transformed in accordance with the predicted pose of the given object, and engages the given object on the basis of the spawned mesh model.

In further examples, a robot can interact with a given object without directly contacting the given object, for example using suction means or blowing means, lasers or other radiation sources, or any other components appropriate to the task performed by the robot.

The system 100 of Figure 1 can be implemented as a standalone device or as a distributed computing system. Although the system 100 includes sensors 106 for capturing images and associated depth information, in other examples a data processing system can be arranged to perform the methods described herein images and associated depth information received from a remote source, in which case sensors are not necessary. In other examples, one or more other types of sensor may be included in addition to, or instead of, the sensors 106. Examples of suitable sensors are stereoscopic cameras, event cameras, infrared cameras, and/or transceivers for sound navigation ranging (sonar).

The processing circuitry 104 of the system 100 includes various processing units including a central processing unit (CPU) and a graphics processing unit (GPU). In other examples, specialist processing units, such as application specific integrated circuits (ASICs) or digital signal processors (DSPs), are provided to perform specific processing operations. In some examples, a specialist neural network accelerator (NNA) or neural processing unit (NPU) is provided for efficiently performing neural network operations. In some examples, a semiconductor device is provided with one or more gate arrays configured to perform specific operations required for the implementation of the methods described herein.

The memory circuitry 102 of the system 100 includes non-volatile storage in the form of a solid-state drive (SSD), along with volatile random-access memory (RAM), in particular static random-access memory (SRAM) and dynamic random-access memory (DRAM). In other examples, alternative types of memory can be included, such as removable storage, flash memory, synchronous DRAM, and so on.

The pose estimation method 200 of Figure 2 or the pose refinement method 700 of Figure 7 can be performed independently of one another. Accordingly, in some examples, a system is configured to perform the method 200 without any further refinement. In such examples, it is not necessary to store volumetric models of each known object, though it is still necessary to provide a trained neural network or other component to recognise the known objects for image segmentation. In other examples, an alternative pose estimation method can be performed in conjunction with the pose refinement method 700.

The above embodiments are to be understood as illustrative examples of the invention. Further embodiments of the invention are envisaged. For example, the pose prediction methods described herein can be combined with physics reasoning to ensure that the resulting pose predictions are physically possible/realistic. Such reasoning can be incorporated using a physics engine, such as are well known in the context of video games. In some examples, physics reasoning is used in addition to, or as an alternative to, collision-based methods for pose refinement.

It is to be understood that any feature described in relation to any one embodiment may be used alone, or in combination with other features described, and may also be used in combination with one or more features of any other of the embodiments, or any combination of any other of the embodiments. Furthermore, equivalents and modifications not described above may also be employed without departing from the scope of the invention, which is defined in the accompanying claims.

CLAIMS

1. A computer-implemented method of estimating a pose of a target object in a three-dimensional scene, the method comprising:
  - 5 obtaining image data and associated depth information representing a view of the three-dimensional scene;
    - processing the image data and the associated depth information to generate a volumetric reconstruction for each of a plurality of objects in the three-dimensional scene, including the target object;
    - 10 determining a volumetric grid containing the target object;
      - generating, using the generated volumetric reconstructions, occupancy data indicating portions of the volumetric grid occupied by free space and portions of the volumetric grid occupied by objects other than the target object; and
      - 15 estimating the pose of the target object using the generated occupancy data and pointwise feature data for a plurality of points on a surface of the target object.
2. The method of claim 1, comprising processing the obtained image data using a two-dimensional feature extractor to generate the pointwise feature data for the plurality of points on the surface of the target object.
- 20 3. The method of claim 2, comprising processing the obtained image data using image segmentation to determine a two-dimensional mask for the target object,
  - wherein processing the obtained image data using the two-dimensional feature extractor comprises processing a portion of the image data corresponding to the two-
  - 25 dimensional mask for the target object.
4. The method of any preceding claim, wherein estimating the pose of the target object using the generated occupancy data and the pointwise feature data comprises:
  - voxelating the pointwise feature data to obtain first volumetric feature data;
  - 30 processing the occupancy data and the first volumetric feature data together using a three-dimensional feature extractor to generate second volumetric feature data;
  - and

estimating the pose of the target object using the second volumetric feature data.

5. The method of any of claim 4, wherein:

the three-dimensional feature extractor is a three-dimensional convolutional  
5 neural network; and

the second volumetric feature data comprises a hierarchy of volumetric features  
generated at respective different layers of the three-dimensional convolutional neural  
network.

10 6. The method of claim 4 or 5, wherein:

the pointwise feature data is first pointwise feature data; and

estimating the pose of the target object using the second volumetric feature data  
comprises:

15 extracting, from the second volumetric feature data, second pointwise  
feature data for the plurality of points on the surface of target object;

determining, using the second pointwise feature data, a candidate pose  
and a corresponding confidence score for each of the plurality of points on the surface  
of the target object; and

20 estimating the pose of the target object from the determined candidate  
poses on the basis of the corresponding confidence scores.

7. The method of claim 6, wherein estimating the pose of the target object from  
the candidate poses comprises determining a candidate pose with a highest  
corresponding confidence score.

25

8. The method of claim 6 or 7, wherein determining the candidate pose and the  
corresponding confidence score for each of the plurality of points on the surface of the  
target object comprises processing the first pointwise feature data and the second  
pointwise feature data together using a regression model.

30

9. The method of any preceding claim, wherein generating the volumetric  
reconstruction for a first object of the plurality of objects comprises:

processing the image data using image segmentation to determine a two-dimensional mask for the first object; and

5 processing a portion of the associated depth information corresponding to the two-dimensional mask for the first object to generate the volumetric reconstruction for the first object.

10. A method of estimating a pose of each of a plurality of target objects in a three-dimensional scene using the method of any preceding claim.

10 11. The method of claim 10, wherein the occupancy data generated for each given target object of the plurality of target objects is first occupancy data, the method comprising, for each given target object of the plurality of target objects:

sampling a plurality of points from a predetermined model of the given target object transformed in accordance with the estimated pose of the given target object;

15 determining respective second occupancy data dependent on positions of the points sampled from the predetermined model of the given target object, relative to the volumetric grid containing the given target object; and

determining respective third occupancy data dependent on positions of the points sampled from the predetermined models of the other target objects of the plurality of target objects, relative to the volumetric grid containing the given target object,

the method further comprising iteratively:

25 determining an occupancy penalty depending on the respective second occupancy data and the respective third occupancy data for each of the plurality of target objects; and

modifying the estimated poses of the plurality of target objects using the determined occupancy penalty.

12. The method of any preceding claim, wherein:

30 the view of the three-dimensional scene is a first view of the three-dimensional scene; and

the estimated pose of the or each target object is a first pose of the given object,

the method comprising:

obtaining further image data and further associated depth information representing a second view of the three-dimensional scene different to the first view of the three-dimensional scene; and

5 for the or each target object:

processing the further image data and the further associated depth information to estimate a second pose for the target object;

transforming at least one of the first pose and the second pose of the target object to determine pose comparison data;

10 processing the pose comparison data to determine whether a consistency condition is met; and

when the consistency condition is determined to be met, generating a predetermined object model for the target object transformed consistently with the first pose and the second pose of the target object.

15

13. An image processing system comprising a one or more sensors operable to capture image data and associated depth information, wherein the system is arranged to:

20 capture, using the one or more sensors, image data and associated depth information representing a view of the three-dimensional scene;

process the image data and the associated depth information to generate a volumetric reconstruction for each of a plurality of objects in the three-dimensional scene, including a target object;

determine a volumetric grid containing the target object;

25 generate, using the generated volumetric reconstructions, occupancy data indicating portions of the volumetric grid occupied by free space and portions of the volumetric grid occupied by objects other than the target object; and

estimate a pose of the target object using the generated occupancy data and pointwise feature data for a plurality of points on a surface of the target object.

30

14. The system of claim 13, comprising a two-dimensional feature extractor,

wherein the system is arranged to process the obtained image data using the two-dimensional feature extractor to generate the pointwise feature data for the plurality of points on the surface of the target object.

5 15. The method of claim 14, wherein:

The system is arranged to process the obtained image data using image segmentation to determine a two-dimensional mask for the target object,

wherein processing the obtained image data using the two-dimensional feature extractor comprises processing a portion of the image data corresponding to the two-  
10 dimensional mask for the target object.

16. The system of any of claims 13 to 15, comprising a three-dimensional feature extractor,

wherein the system is arranged to estimate the pose of the target object using  
15 the generated occupancy data and the pointwise feature data by:

voxelating the pointwise feature data to obtain first volumetric feature data;

processing the occupancy data and the first volumetric feature data together using the three-dimensional feature extractor to generate second volumetric feature data; and

20 estimating the pose of the target object using the second volumetric feature data.

17. The system of claim 16, wherein:

the three-dimensional feature extractor is a three-dimensional convolutional neural network arranged to generate the second volumetric feature data; and

25 the second volumetric feature data comprises volumetric features generated at respective different layers of the three-dimensional convolutional neural network.

18. The system of claim 16 or 17, wherein:

the pointwise feature data is first pointwise feature data; and

30 the system is arranged to estimate the pose of the target object using the second volumetric feature data by:

extracting, from the second volumetric feature data, second pointwise feature data for the plurality of points on the surface of the target object;

determining, using the second pointwise feature data, a candidate pose and a corresponding confidence score for each of the plurality of points on the surface  
5 of the target object; and

estimating the pose of the target object from the determined candidate poses on the basis of the corresponding confidence scores.

19. The system of claim 18, arranged to estimate the pose of the target object from  
10 the candidate poses by determining a candidate pose with a highest corresponding confidence score.

20. The system of claim 18 or 19, arranged to determine the candidate pose and the corresponding confidence score for each of the plurality of points on the surface of the  
15 target object by processing the first pointwise feature data and the second pointwise feature data together using a regression model.

21. The system of any of claims 13 to 20, arranged to generate the volumetric reconstruction for a first object of the plurality of objects by:

20 processing the image data using image segmentation to determine a two-dimensional mask for the first object; and

processing a portion of the associated depth information corresponding to the two-dimensional mask for the first object to generate the volumetric reconstruction for the first object.  
25

22. The system of any of claims 13 to 21, comprising engaging means for engaging the target object in dependence on the estimated pose of the target object.

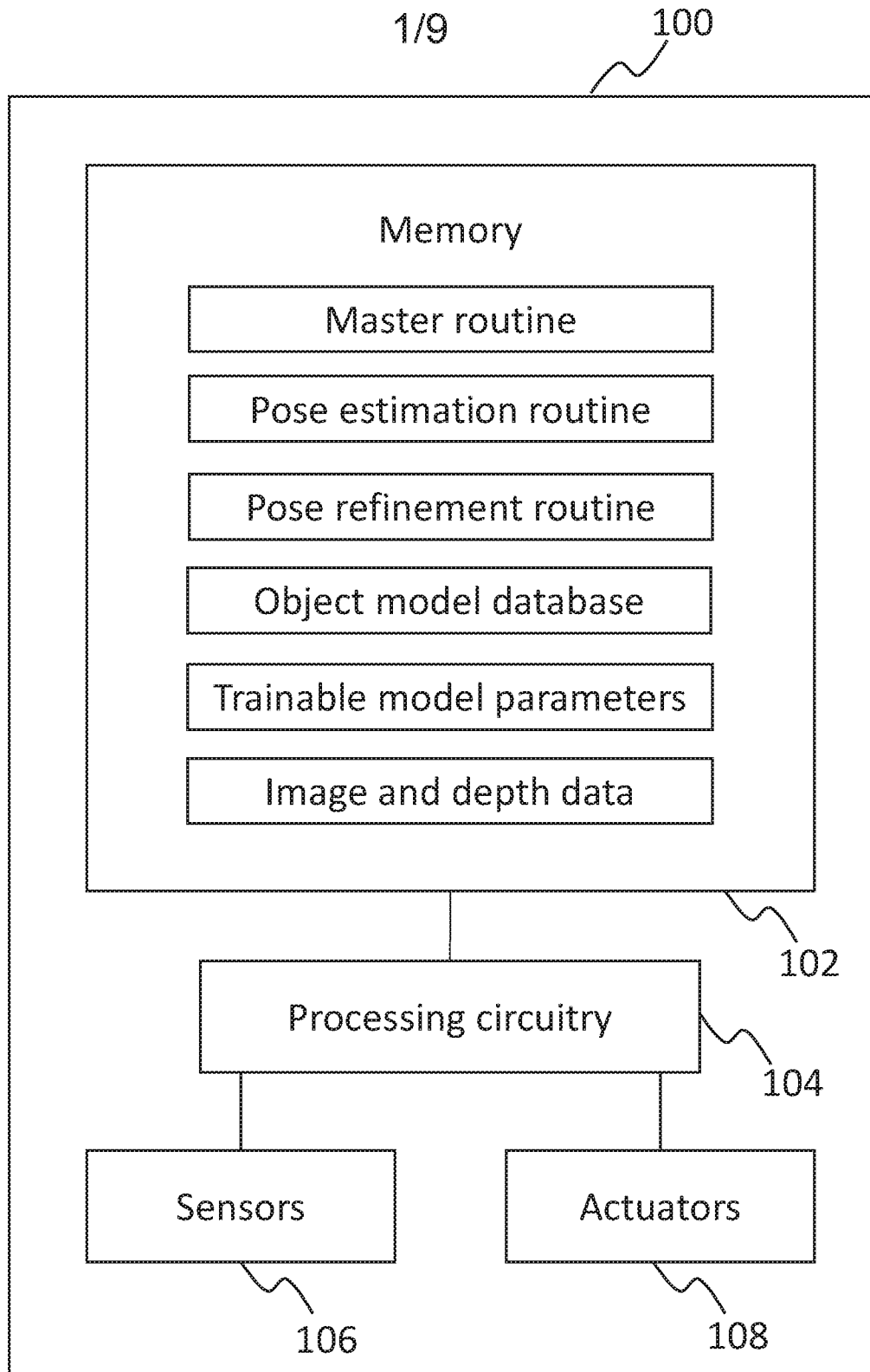
23. The system of any of claims 13 to 22, comprising one or more actuators  
30 arranged to move the one or more sensors, wherein:

the view of the three-dimensional scene is a first view of the three-dimensional scene;

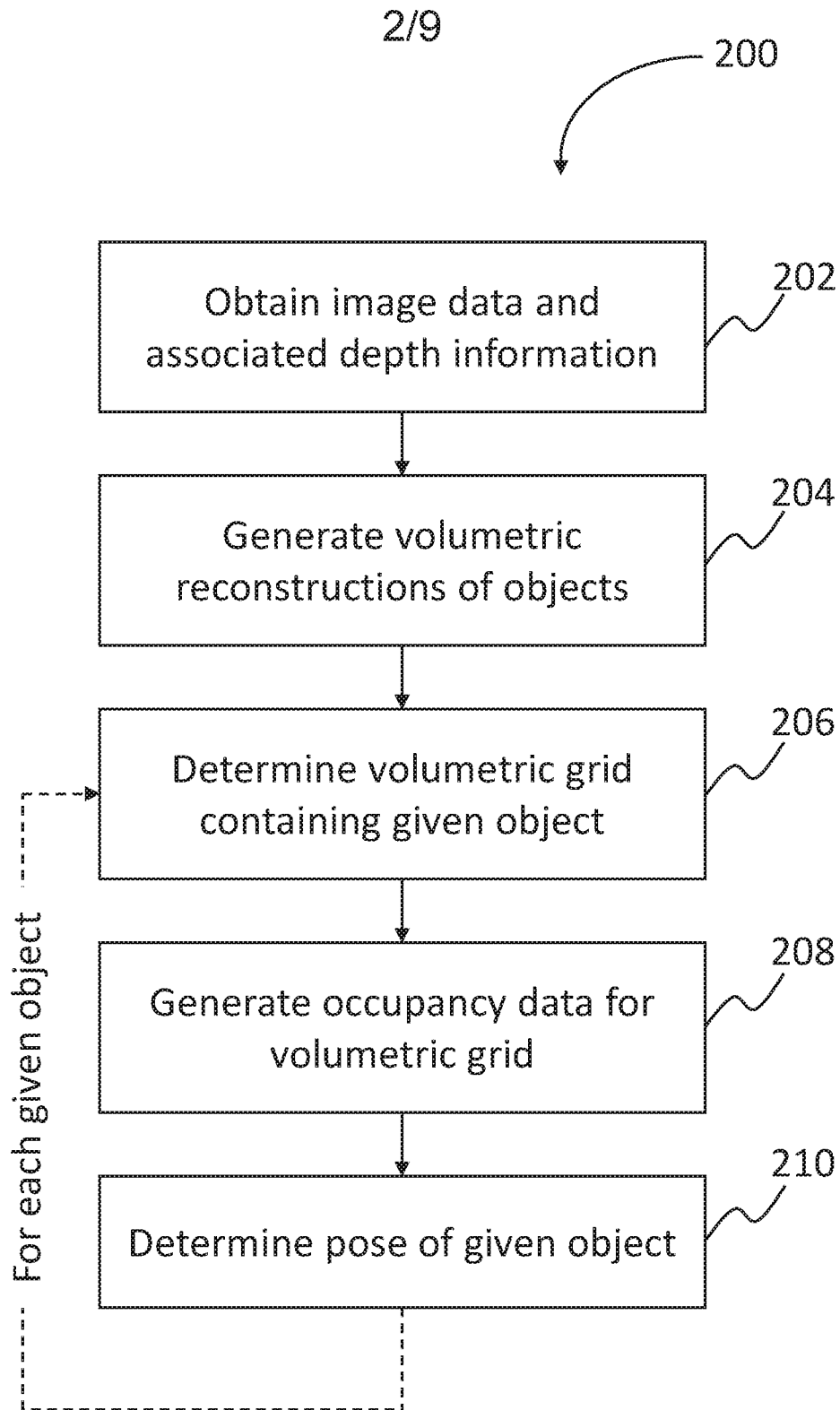
the determined pose of the target object is a first pose; and  
the system is arranged to:

- move the one or more sensors using the one or more actuators;
- capture further image data and further associated depth information
- 5 representing a second view of the three-dimensional scene resulting from the moving  
of the one or more sensors;
- process the further image data and the further associated depth  
information to estimate a second pose for the target object;
- transform at least one of the first pose and the second pose for the given
- 10 object to generate pose comparison data;
- process the pose comparison data to determine whether a consistency  
condition is met; and
- when the consistency condition is determined to be met, generate a  
predetermined object model for the target object transformed in accordance with the
- 15 determined first pose and the determined second pose.

24. A computer program product comprising machine-readable instructions which,  
when executed by a computing system, cause the computing system to perform the  
method of any of claims 1 to 12.

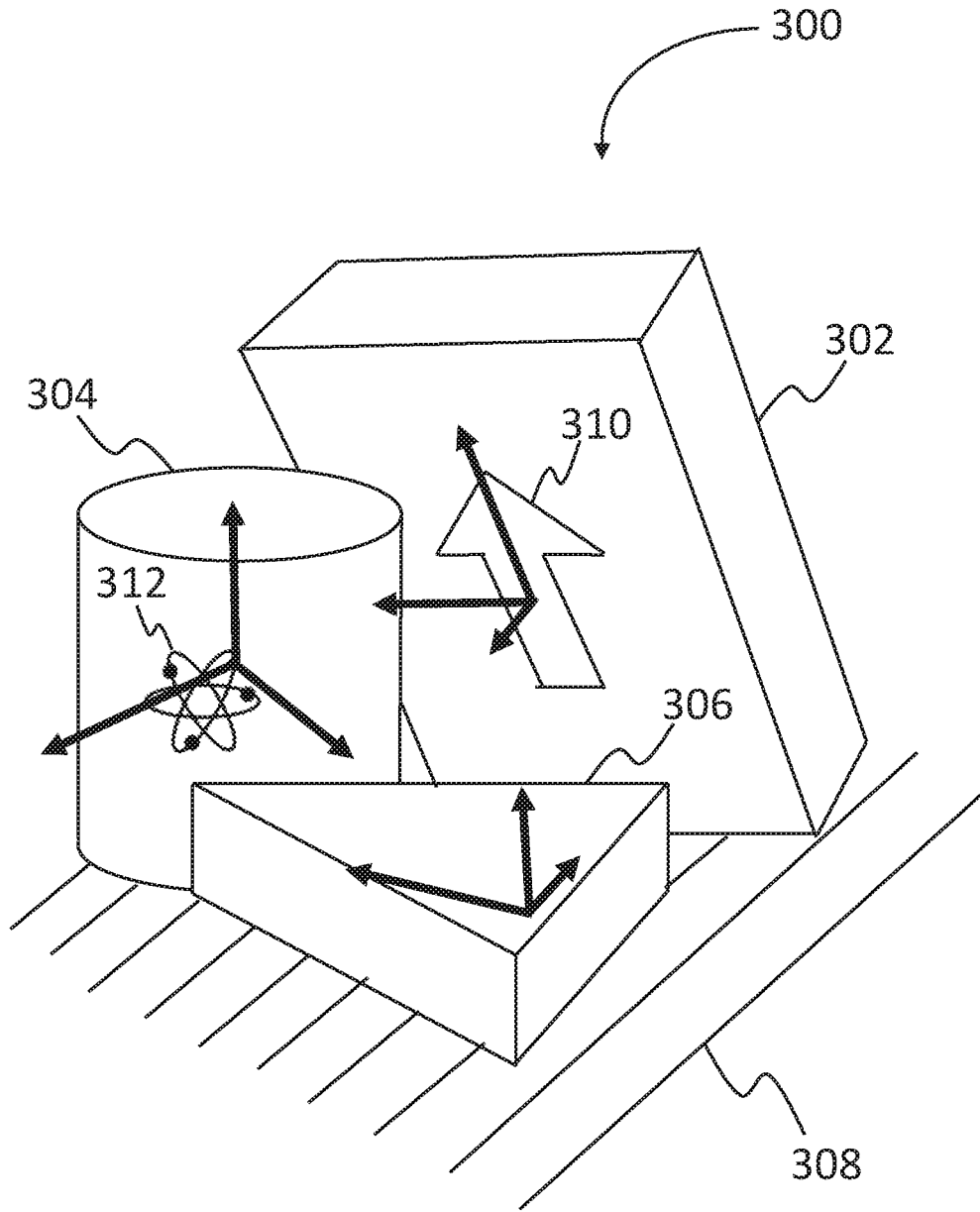


**Fig. 1**

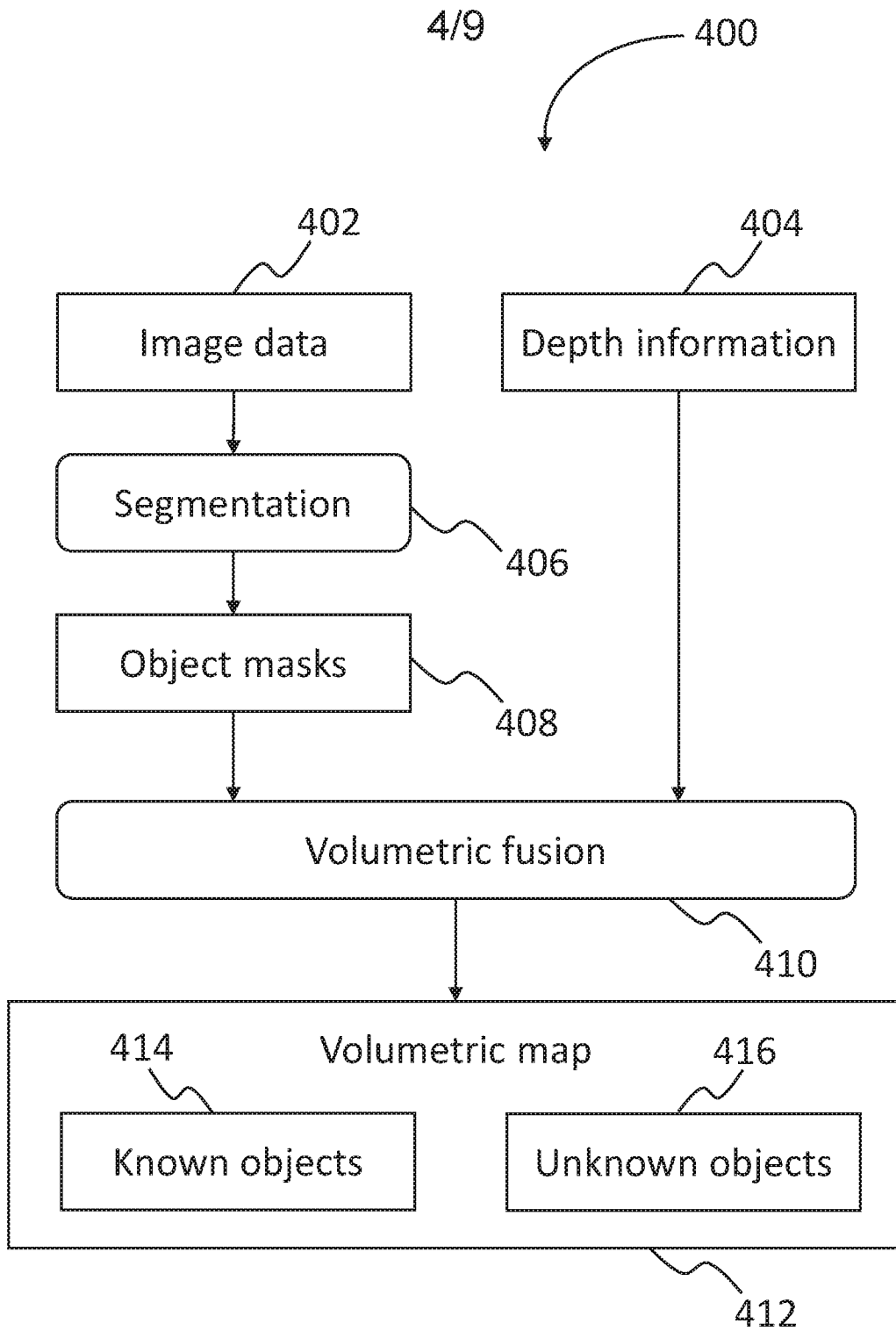


**Fig. 2**

3/9

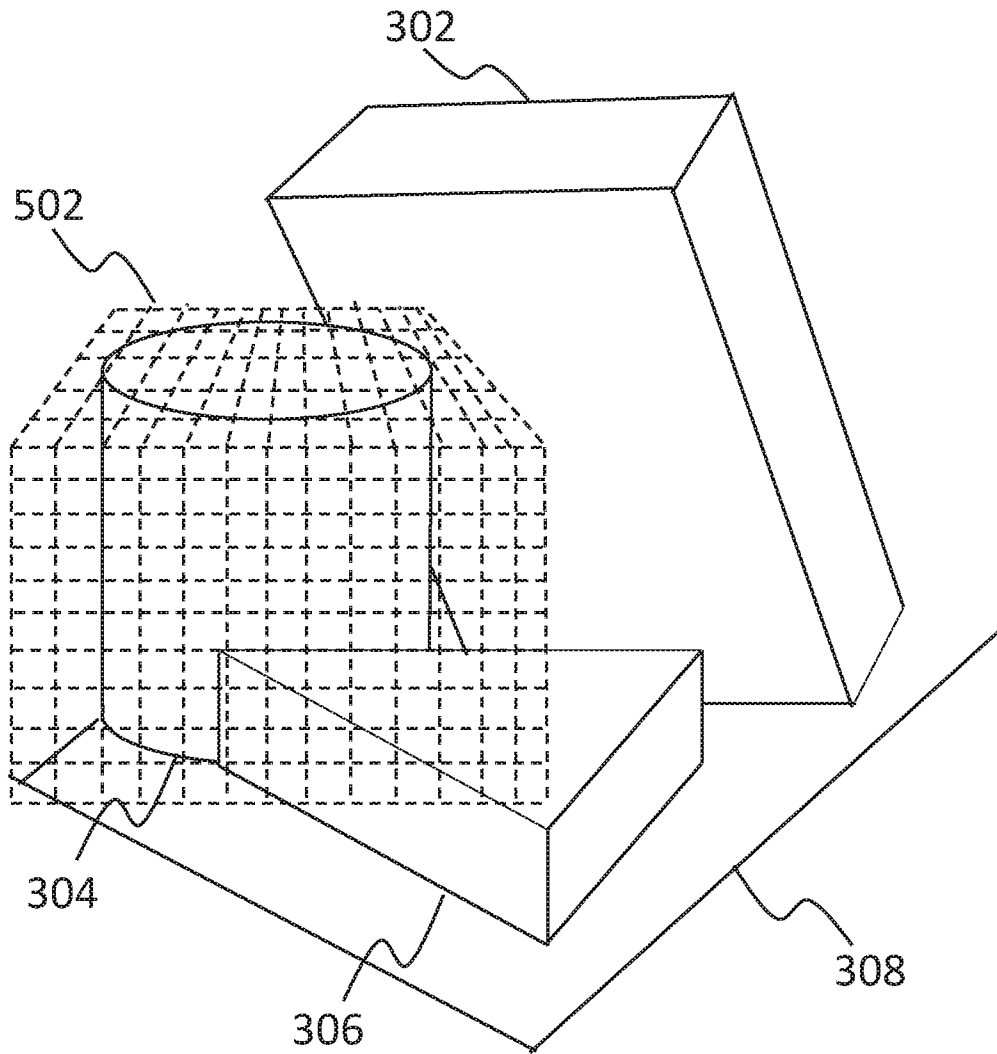


**Fig. 3**

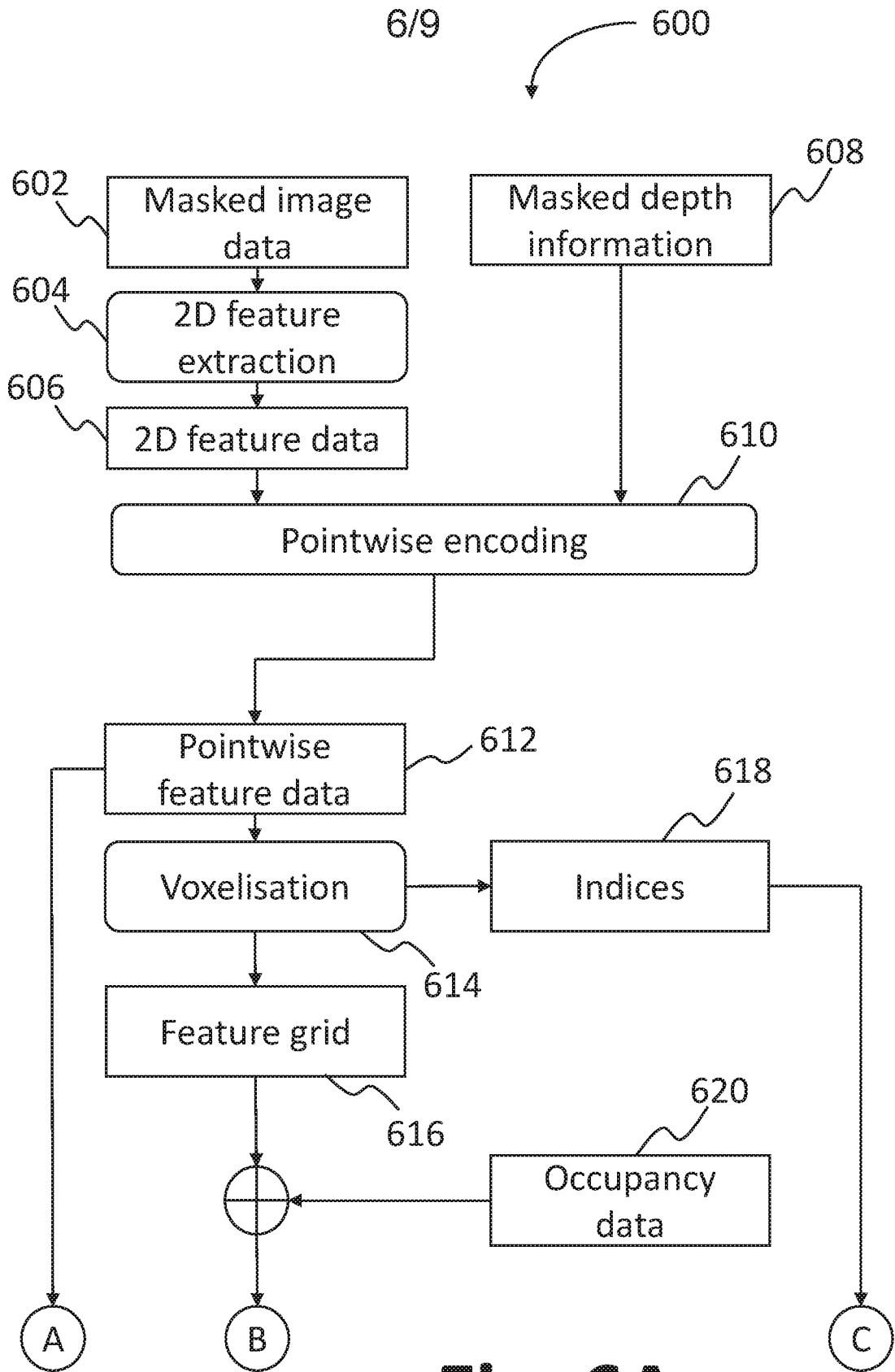


**Fig. 4**

5/9

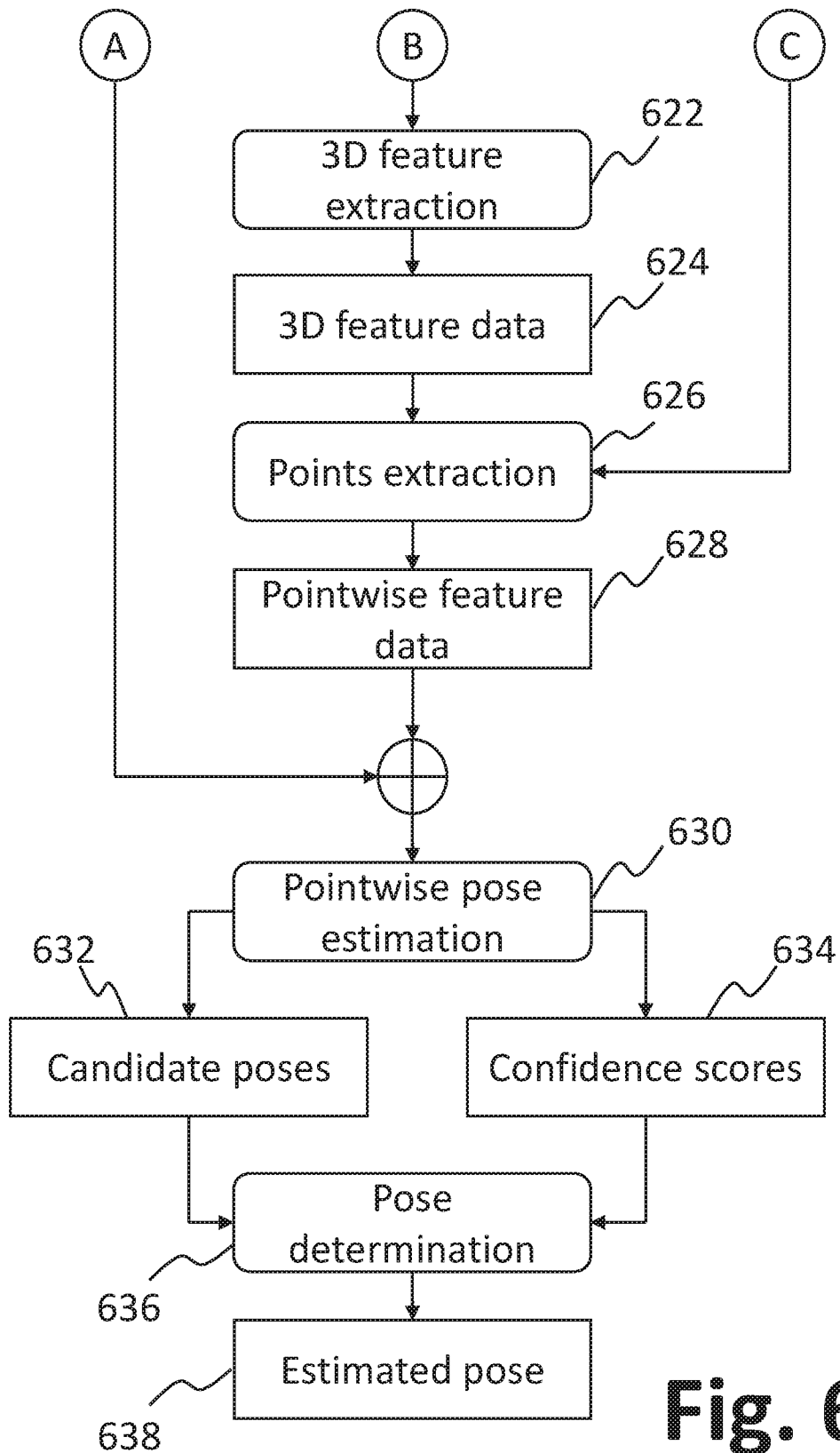


**Fig. 5**

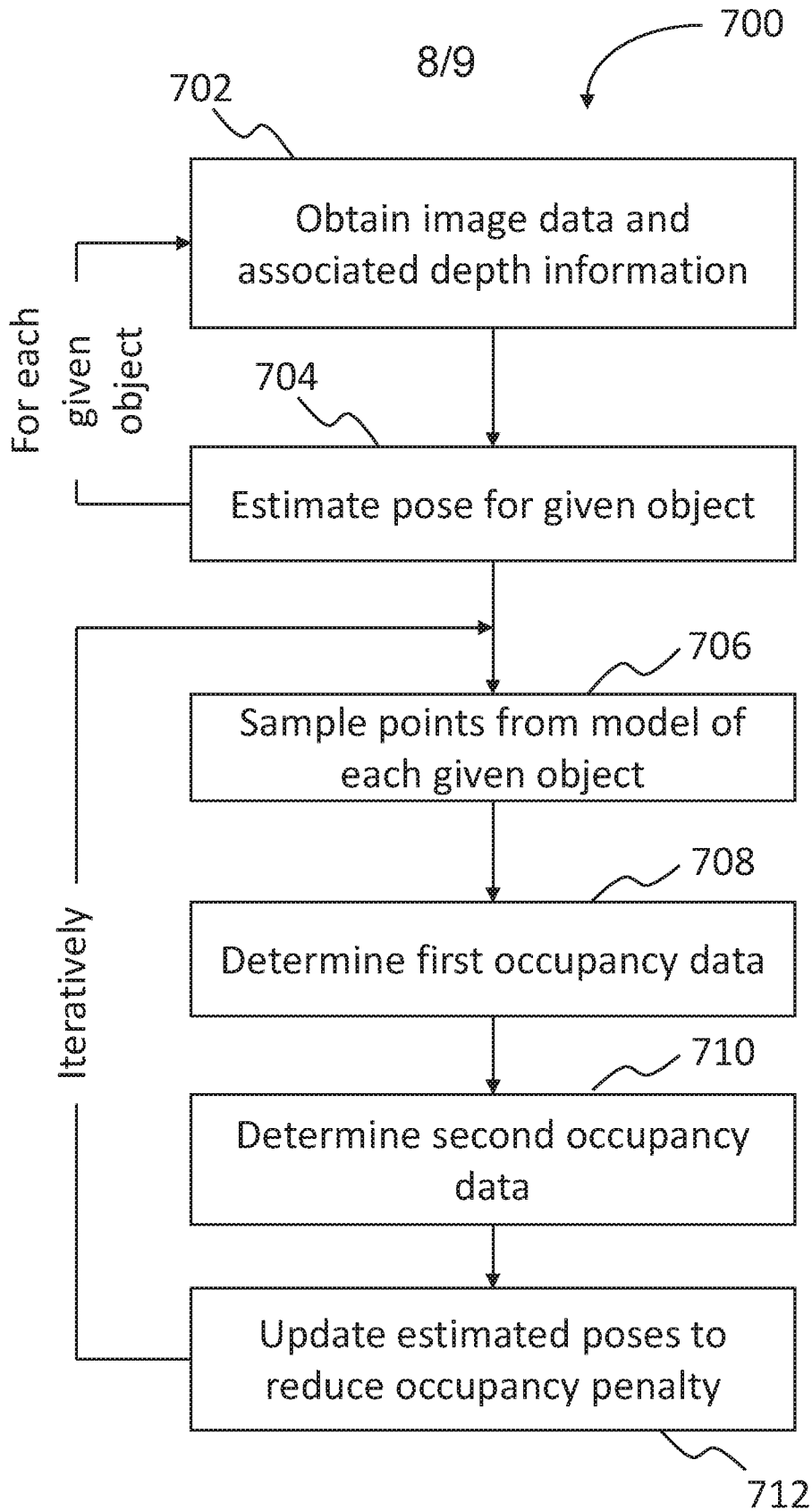


**Fig. 6A**

7/9

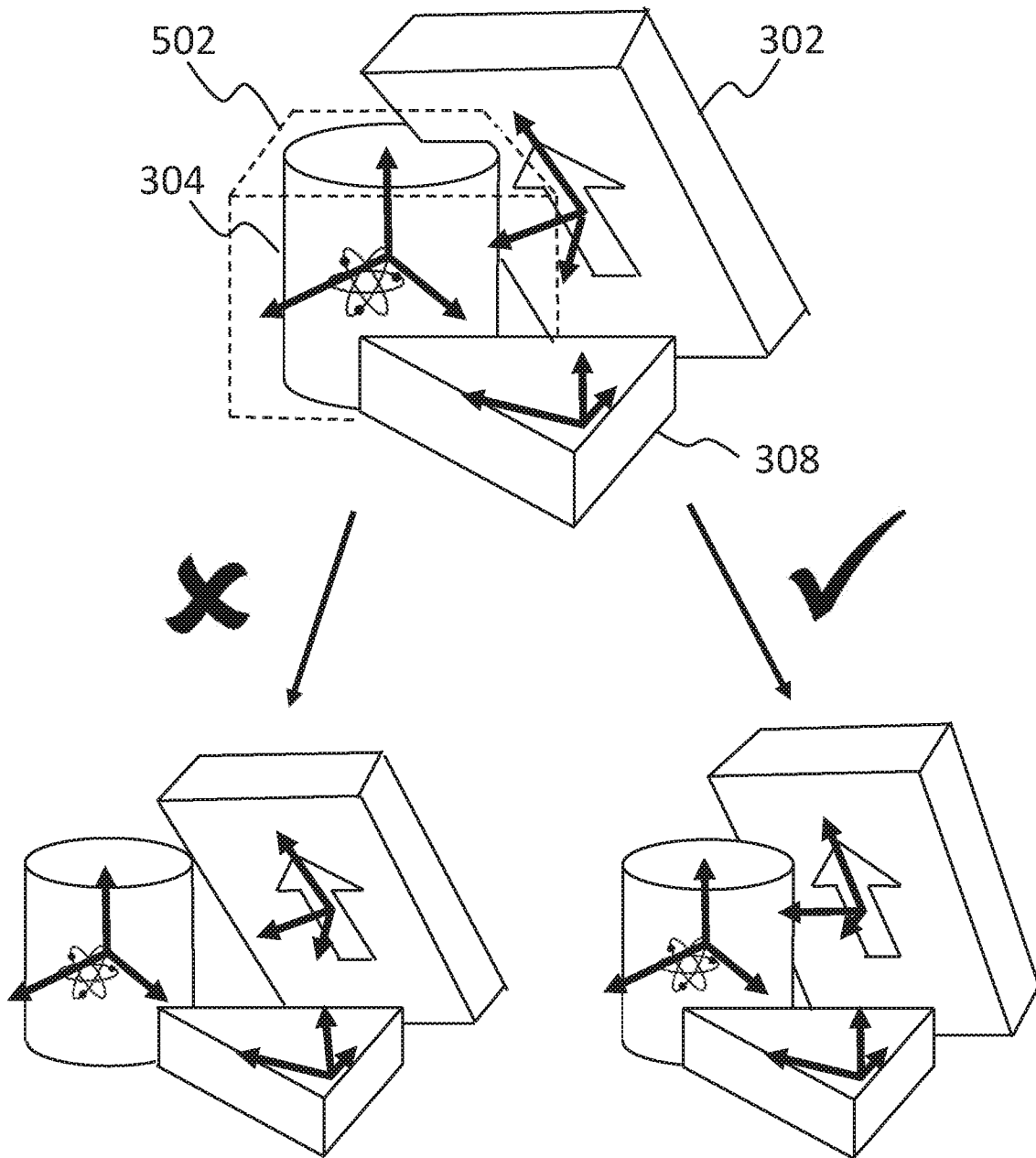


**Fig. 6B**



**Fig. 7**

9/9



**Fig. 8**

**INTERNATIONAL SEARCH REPORT**

International application No  
PCT/GB2021/050770

A. CLASSIFICATION OF SUBJECT MATTER  
 INV. G06T7/00 G06T7/11 G06T7/70 G06T7/73 G06T17/00  
 ADD.  
 According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED  
 Minimum documentation searched (classification system followed by classification symbols)  
 G06K G06T  
 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WANG CHEN ET AL: "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion", 2019 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), IEEE, 15 June 2019 (2019-06-15), pages 3338-3347, XP033686474, DOI: 10.1109/CVPR.2019.00346 [retrieved on 2020-01-08] abstract; figure 2 section: 3.3. Dense Feature Extraction ----- -/--	1-24

Further documents are listed in the continuation of Box C.

See patent family annex.

\* Special categories of cited documents :

<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&amp;" document member of the same patent family</p>
---	---

Date of the actual completion of the international search <b>22 June 2021</b>	Date of mailing of the international search report <b>30/06/2021</b>
--	---

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

## INTERNATIONAL SEARCH REPORT

International application No  
PCT/GB2021/050770

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>JOHN MCCORMAC ET AL: "Fusion++: Volumetric Object-Level SLAM", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 25 August 2018 (2018-08-25), XP081266011, abstract</p> <p style="text-align: center;">-----</p>	1-24
A	<p>XU BINBIN ET AL: "MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM", 2019 INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), IEEE, 20 May 2019 (2019-05-20), pages 5231-5237, XP033594350, DOI: 10.1109/ICRA.2019.8794371 [retrieved on 2019-08-09] abstract</p> <p style="text-align: center;">-----</p>	1-24
A	<p>GB 2 576 322 A (IMPERIAL COLLEGE SCI TECH &amp; MEDICINE [GB]) 19 February 2020 (2020-02-19) abstract</p> <p style="text-align: center;">-----</p>	1,13

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2021/050770

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB 2576322	A	19-02-2020	
		CN 112602116 A	02-04-2021
		EP 3837667 A1	23-06-2021
		GB 2576322 A	19-02-2020
		KR 20210042942 A	20-04-2021
		SG 11202100469R A	25-02-2021
		TW 202034215 A	16-09-2020
		US 2021166426 A1	03-06-2021
		WO 2020035661 A1	20-02-2020
-----			