

(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2020331567 B2**

(54) Title
Object moving system

(51) International Patent Classification(s)
H04N 5/265 (2006.01) **H04N 9/47** (2006.01)
H04N 7/18 (2006.01) **H04N 13/204** (2018.01)

(21) Application No: **2020331567** (22) Date of Filing: **2020.09.10**

(30) Priority Data

(31) Number	(32) Date	(33) Country
2019903398	2019.09.12	AU
PCT/AU2020/050113	2020.02.11	AU

(43) Publication Date: **2021.04.15**

(43) Publication Journal Date: **2021.04.15**

(44) Accepted Journal Date: **2022.02.03**

(71) Applicant(s)
Commonwealth Scientific and Industrial Research Organisation

(72) Inventor(s)
FLICK, Paul;PANITZ, Nic;DEAN, Peter;ELMOUTTIE, Marc;LIANG, Sisi;STEINDL, Ryan;CORDIE, Troy;BANDYOPADHYAY, Tirtha

(74) Agent / Attorney
Davies Collison Cave Pty Ltd, Level 10 301 Coronation Drive, MILTON, QLD, 4064, AU

(56) Related Art
WO 2019005727 A1
WO 2018119450 A1
US 2017/0313332 A1



(51) International Patent Classification:

G05D 1/08 (2006.01) G05B 19/00 (2006.01)
G05D 1/02 (2020.01)

(21) International Application Number:

PCT/AU2020/050961

(22) International Filing Date:

10 September 2020 (10.09.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2019903398 12 September 2019 (12.09.2019) AU
PCT/AU2020/050113
11 February 2020 (11.02.2020) AU

(71) Applicant: **COMMONWEALTH SCIENTIFIC AND INDUSTRIAL RESEARCH ORGANISATION** [AU/AU]; Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU).

(72) Inventors: **FLICK, Paul**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU). **PANITZ, Nic**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU). **DEAN, Peter**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU).

ELMOUTTIE, Marc; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU). **LIANG, Sisi**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU). **STEINDL, Ryan**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU). **CORDIE, Troy**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU). **BANDY-OPADHYAY, Tirtha**; c/o Commonwealth Scientific and Industrial Research Organisation, Clunies Ross Street, Acton, Australian Capital Territory 2601 (AU).

(74) Agent: **DAVIES COLLISON CAVE PTY LTD**; Level 10, 301 Coronation Drive, Milton, Queensland 4064 (AU).

(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.

(54) Title: OBJECT MOVING SYSTEM

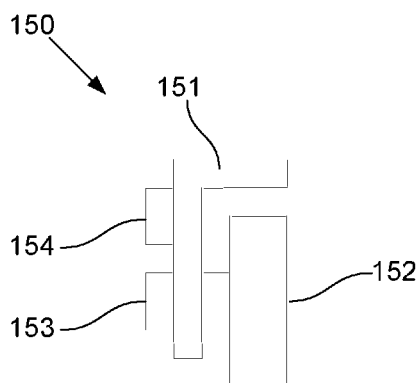


Fig. 1A

(57) Abstract: A system for moving an object within an environment, wherein the system includes at least one modular wheel configured to move the object. The modular wheel includes a body configured to be attached to the object, a wheel, a drive configured to rotate the wheel and a controller configured to control the drive. One or more processing devices configured are provided to receive an image stream including a plurality of captured images from each of a plurality of imaging devices, the plurality of imaging devices being configured to capture images of the object within the environment, analyse the images to determine an object location within the environment, generate control instructions at least in part using the determined object location and provide the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.

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

Published:

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

OBJECT MOVING SYSTEM

Background of the Invention

[0001] The present invention relates to a system and method for moving an object in within an environment, and in one particular example, to a system and method for moving an object using one or more modular wheels attached to the object.

Description of the Prior Art

[0002] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that the prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

[0003] “Enabling rapid field deployments using modular mobility units” by Troy Cordie, Tirthankar Bandyopadhyay, Jonathan Roberts, Ryan Steindl, Ross Dungavell and Kelly Greenop, Australasian Conference on Robotics and Automation 2016, ACRA 2016 P107-115, describes a set of modular wheels that enable bespoke platform development for rapid field deployments. The modular wheel is influenced by existing modular and inspection robots but provides a simple-to-operate solution to exploring various environments. Each wheel provides two degrees of freedom allowing any continuous orientation to be achieved within a plane. Onboard computing and a wi-fi connection enable the modular wheels to operate individually or collaboratively. Heterogeneous robot platforms can be created as required through the use of adaptors. With robots of differing shapes, sizes and configurations able to be created at run time as demonstrated within the laboratory and in the field. The dynamic nature of the system model dictates the control characteristics providing differential, Ackerman and nonholonomic omnidirectional control options to the user.

[0004] In the above described system, overall control is performed by an operator, thereby limiting the system to use with manual control.

Summary of the Present Invention

[0005] In one broad form, an aspect of the present invention seeks to provide a system for moving an object within an environment, wherein the system includes: at least one modular wheel configured to move the object, wherein the at least one modular wheel includes: a body configured to be attached to the object; a wheel; a drive configured to rotate the wheel; and, a controller configured to be attached to the body and to control the drive; and, one or more processing devices configured to: receive an image stream including a plurality of captured images from each of a plurality of imaging devices positioned in the environment, the plurality of imaging devices being configured to capture images of the object within the environment; analyse the images to determine an object location within the environment; generate control instructions at least in part using the determined object location; and, provide the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.

[0006] In one embodiment the system includes one or more passive wheels mounted to the object.

[0007] In one embodiment the at least one modular wheel includes a steering drive configured to adjust an orientation of the wheel, and wherein the controller is configured to control the steering drive to thereby change an orientation of the wheel.

[0008] In one embodiment the at least one modular wheel includes a transceiver configured to communicate wirelessly with the one or more processing devices.

[0009] In one embodiment the at least one modular wheel includes a power supply configured to power at least one of: the drive; the controller; a transceiver; and, a steering drive.

[0010] In one embodiment the control instructions include at least one of: a wheel orientation for each wheel; and, a rate of rotation for each wheel.

[0011] In one embodiment the system includes a plurality of modular wheels.

- 3 -

[0012] In one embodiment the one or more processing devices are configured to provide respective control instructions to each controller to thereby independently control each modular wheel.

[0013] In one embodiment the one or more processing devices are configured to provide control instructions to the controllers and wherein the controllers communicate to independently control each modular wheel.

[0014] In one embodiment the control instructions include a direction and rate of travel for the object, and wherein the controllers use the control instructions to determine at least one of: a wheel orientation for each wheel; and, a rate of rotation for each wheel.

[0015] In one embodiment the system is configured to steer the vehicle by at least one of: differentially rotating multiple modular wheels; and, changing an orientation of one or more modular wheels.

[0016] In one embodiment the one or more processing devices are configured to: determine an object configuration; and, generate the control instructions at least partially in accordance with the object extent.

[0017] In one embodiment the object configuration is indicative of at least one of: a physical extent of the object; and, movement parameters associated with the object.

[0018] In one embodiment the one or more processing devices are configured to: determine a wheel configuration indicative of a position of each wheel relative to the object; and, generate the control instructions at least partially in accordance with the wheel configuration.

[0019] In one embodiment the at least one modular wheel are each attached to the object at known locations.

[0020] In one embodiment the object includes a platform and wherein the at least one modular wheel is attached to the platform.

[0021] In one embodiment the object includes an item supported by the platform.

[0022] In one embodiment the one or more processing devices are configured to: determine an identity of at least one of: each modular wheel; and, the object; and, generate control instructions in accordance with the identity. A system according to claim 18, wherein the one or more processing devices are configured to determine the identity at least in part using a network identifier.

[0023] In one embodiment the one or more processing devices are configured to determine the identity using machine readable coded data.

[0024] In one embodiment the machine readable coded data is visible data, and wherein the one or more processing devices are configured to analyse the images to detect the machine readable coded data.

[0025] In one embodiment the machine readable coded data is encoded on a tag, and wherein the one or more processing devices are configured to receive signals indicative of the machine readable coded data from a tag reader.

[0026] In one embodiment the tags at least one of: short range wireless communications protocol tags; RFID tags; and, Bluetooth tags.

[0027] In one embodiment the one or more processing devices are configured to: determine routing data indicative of at least one of: a travel path; and, a destination; generate control instructions in accordance with the routing data and the object location.

[0028] In one embodiment the routing data is indicative of at least one of: a permitted object travel path; permitted object movements; permitted proximity limits for different objects; permitted zones for objects; denied zones for objects.

[0029] In one embodiment the one or more processing devices are configured to: determine an identity for at least one of: the object; and, for at least one modular wheel attached to the object; determine the routing data at least in part using the object identity.

[0030] In one embodiment the one or more processing devices are configured to: generate calibration control instructions; monitor movement of at least one of the object and the at

least one modular wheel in response to the calibration control instructions; and use results of the monitoring to generate control instructions.

[0031] In one embodiment the imaging devices are at least one of: positioned within the environment at fixed locations; and, static relative to the environment.

[0032] In one embodiment at least some of the imaging devices are positioned within the environment to have at least partially overlapping fields of view and wherein the one or more processing devices are configured to: identify overlapping images in the different image streams, the overlapping images being images captured by imaging devices having overlapping fields of view; and, analyse the overlapping images to determine object locations within the environment.

[0033] In one embodiment at least some of the imaging devices are positioned within the environment to have at least partially overlapping fields of view and wherein the one or more processing devices are configured to: analyse changes in the object locations over time to determine object movements within the environment; compare the object movements to situational awareness rules; and, use results of the comparison to identify situational awareness events.

[0034] In one embodiment the overlapping images are synchronous overlapping images captured at approximately the same time.

[0035] In one embodiment the one or more processing devices are configured to: determine a capture time of each captured image; and, identify synchronous images using the captured time.

[0036] In one embodiment the one or more processing devices are configured to determine a capture time using at least one: a capture time generated by the imaging device; a receipt time associated with each image, the receipt time being indicative of a time of receipt by the one or more processing devices; and, a comparison of image content in the images.

[0037] In one embodiment the one or more processing devices are configured to: analyse images from each image stream to identify object images, the object images being images

including objects; and, identify overlapping images as object images that include the same object.

[0038] In one embodiment the one or more processing devices are configured to identify overlapping images based at least in part on a positioning of the imaging devices.

[0039] In one embodiment the one or more processing devices are configured to: analyse a number of images from an image stream to identify static image regions; and, identifying object images as images including non-static image regions.

[0040] In one embodiment at least one of the images is a background reference image.

[0041] In one embodiment the one or more processing devices are configured to determine an object location using at least one of: a visual hull technique; and, detection of fiducial markings in the images; and, detection of fiducial markings in multiple triangulated images.

[0042] In one embodiment the one or more processing devices are configured to interpret the images in accordance with calibration data.

[0043] In one embodiment the calibration data includes at least one of: intrinsic calibration data indicative of imaging properties of each imaging device; and, extrinsic calibration data indicative of relative positioning of the imaging devices within the environment.

[0044] In one embodiment the one or more processing devices are configured to generate calibration data during a calibration process by: receiving images of defined patterns captured from different positions using an imaging device; and, analysing the images to generate calibration data indicative of a image capture properties of the imaging device.

[0045] In one embodiment the one or more processing devices are configured to generate calibration data during a calibration process by: receiving captured images of targets within the environment; analysing the captured images to identify images captured by different imaging devices which show the same target; and, analysing the identified images to generate calibration data indicative of a relative position and orientation of the imaging devices.

[0046] In one embodiment the one or more processing devices are configured to generate an environment model, the environment model being indicative of at least one of: the environment; a location of imaging devices in the environment; current object locations; object movements; predicted obstacles; predicted object locations; and, predicted object movements.

[0047] In one embodiment the one or more processing devices are configured to generate a graphical representation of the environment model.

[0048] In one embodiment the one or more processing devices are configured to: analyse changes in object locations over time to determine object movements within the environment; compare the object movements to situational awareness rules; and, use results of the comparison to identify situational awareness events.

[0049] In one embodiment in response to identification of a situational awareness event, the one or more processing devices are configured to perform an action including at least one of: record an indication of the situational awareness event; generate a notification indicative of the situational awareness event; cause an output device to generate an output indicative of the situational awareness event; activate an alarm; and, cause operation of an object to be controlled.

[0050] In one embodiment the one or more processing devices are configured to: identify the situational awareness event substantially in real time; and, perform an action substantially in real time.

[0051] In one embodiment the imaging devices are at least one of: security imaging devices; monoscopic imaging devices; non-computer vision based imaging devices; and, imaging devices that do not have associated intrinsic calibration information.

[0052] In one broad form, an aspect of the present invention seeks to provide a method for moving an object within an environment, the method being performed using a system including: at least one modular wheel configured to move the object, wherein the at least one modular wheel each includes: a body configured to be attached to the object; a wheel; a drive configured to rotate the wheel; and, a controller configured to control the drive; and, one or

processing devices, wherein the method includes, in the one or more processing devices: receiving an image stream including a plurality of captured images from each of a plurality of imaging devices positioned in the environment, the plurality of imaging devices being configured to capture images of the object within the environment; analysing the images to determine an object location within the environment; generating control instructions at least in part using the determined object location; and, providing the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.

[0053] In one broad form, an aspect of the present invention seeks to provide a computer program product for moving an object within an environment using a system including: at least one modular wheel configured to move the object, wherein the at least one modular wheel each includes: a body configured to be attached to the object; a wheel; a drive configured to rotate the wheel; and, a controller configured to control the drive; and, one or more processing devices, wherein the computer program product includes computer executable code, which when executed by the one or more processing devices causes the one or more processing devices to: receive an image stream including a plurality of captured images from each of a plurality of imaging devices positioned in the environment, the plurality of imaging devices being configured to capture images of the object within the environment; analyse the images to determine an object location within the environment; generate control instructions at least in part using the determined object location; and, provide the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.

[0054] It will be appreciated that the broad forms of the invention and their respective features can be used in conjunction and/or independently, and reference to separate broad forms is not intended to be limiting. Furthermore, it will be appreciated that features of the method can be performed using the system or apparatus and that features of the system or apparatus can be implemented using the method.

Brief Description of the Drawings

[0055] Various examples and embodiments of the present invention will now be described with reference to the accompanying drawings, in which: -

[0056] Figure 1A is a schematic end view of an example of a modular wheel;

[0057] Figure 1B is a schematic side view of the modular wheel of Figure 1A;

[0058] Figure 1C is a schematic end view of modular wheels of Figure 1A mounted to an object;

[0059] Figure 1D is a schematic side view of the object of Figure 1C;

[0060] Figure 2 is a schematic diagram of an example of a system for object monitoring within an environment;

[0061] Figure 3 is a flowchart of an example of a method for moving an object within an environment;

[0062] Figure 4 is a schematic diagram of an example of a distributed computer system;

[0063] Figure 5 is a schematic diagram of an example of a processing system;

[0064] Figure 6 is a schematic diagram of an example of a client device;

[0065] Figure 7A is a schematic end view of a specific example of a modular wheel;

[0066] Figure 7B is a schematic side view of the modular wheel of Figure 7A;

[0067] Figure 8 is a schematic diagram of an example of a wheel controller for the modular wheel of Figure 7A;

[0068] Figures 9A to 9D are schematic diagrams of examples of different wheel control configurations;

[0069] Figure 10 is a flowchart of an example of a method for moving an object within an environment;

- 10 -

[0070] Figures 11A and 11B are a flowchart of a further example of a method for moving an object within an environment;

[0071] Figure 12 is a flowchart of an example of a calibration method for use with a system for moving an object within an environment;

[0072] Figures 13A to 13C are a flowchart of a specific example of a method for moving an object within an environment;

[0073] Figure 14 is a flowchart of an example of a method of identifying objects as part of a situational awareness monitoring method;

[0074] Figure 15 is a schematic diagram of an example of a graphical representation of a situational awareness model;

[0075] Figure 16 is a flow chart of an example of a process for image region classification;

[0076] Figure 17 is a flow chart of an example of a process for occlusion mitigation; and,

[0077] Figure 18 is a flow chart of an example of a process for weighted object detection.

Detailed Description of the Preferred Embodiments

[0078] An example of a modular wheel for moving an object within an environment will now be described with reference to Figures 1A to 1D.

[0079] In this example, the modular wheel 150 includes a body 151 configured to be attached to the object. The body could be of any appropriate form and could be attached to the object in any manner, including through the use of a mounting bracket or similar.

[0080] The modular wheel includes a wheel 152, typically supported by the body using an axle or similar, and a drive 153, such as a motor, configured to rotate the wheel. A controller 154 is provided which is configured to control the drive 153, and thereby allow the wheel 152 to be rotated as required.

[0081] The controller could be of any appropriate form but in one example is a processing system that executes software applications stored on non-volatile (e.g., hard disk) storage, although this is not essential. However, it will also be understood that the controller could be any electronic processing device such as a microprocessor, microchip processor, logic gate configuration, firmware optionally associated with implementing logic such as an FPGA (Field Programmable Gate Array), or any other electronic device, system or arrangement.

[0082] In use, one or more modular wheels can be attached to an object to allow the object to be moved, and an example of this will now be described with reference to Figures 1C and 1D.

[0083] In this example, an object 160 in the form of a platform is shown, with four modular wheels 150 being mounted to the platform, allowing the platform to be moved by controlling each of the four modular wheels 150. However, a wide range of different arrangements are contemplated, and the above example is for the purpose of illustration only and is not intended to be limiting.

[0084] For example, the system could use a combination of driven modular wheels and passive wheels, with the one or more modular wheels could be used to provide motive force, whilst passive wheels are used to fully support the object. Steering could be achieved by steering individual wheels, as will be described in more detail below and/or through differential rotation of different modular wheels, for example using skid steer arrangements, or similar.

[0085] In the current example, modular wheels are shown provided proximate corners of the platform. However, this is not essential and the modular wheels could be mounted at any location, assuming this is sufficient to adequately support the platform.

[0086] Whilst the current example focuses on the use of a platform, the modular wheel could be used with a wide range of different objects. For example, using the wheels with platforms, pallets, or other similar structures, allows one or more items to be supported by the platform, and moved collectively. Thus, wheels could be attached to a pallet supporting a number of items, allowing the pallet and items to be moved without requiring the use of a

pallet jack or similar. In this instance, the term object is intended to refer collectively to the platform/pallet and any items supported thereon. Alternatively, the wheels could be attached directly to an item, without requiring a platform, in which case the item is the object.

[0087] The nature of objects that can be moved will vary depending on the preferred implementation, intended usage scenario, and the nature of the environment. Particular example environments include factories, warehouses, storage environments, or similar, although it will be appreciated that the techniques could be applied more broadly, and could be used in indoor and/or outdoor environments. Similarly, the objects could be a wide variety of objects, and may for example include items to be moved within a factory, such as components of vehicles, or the like. However, it will be appreciated that this is not intended to be limiting.

[0088] A system for controlling movement of objects within an environment will now be described with reference to Figure 2.

[0089] In this example, movement control is performed within an environment *E* where a movable object and optionally one or more other objects 201, 202, 203 are present. In this regard, the term movable object 160 refers to any object that is movable by virtue of having attached modular wheels, whilst the term other object 201, 202, 203 refers to object that do not include modular wheels, and which can be static objects and/or objects that move using mechanisms other than modular wheels. Such other objects could be a wide variety of objects, and in particular moving objects, such as people, animals, vehicles, autonomous or semiautonomous vehicles, such as automated guided vehicles (AGVs), or the like, although this is not intended to be limiting. Whilst four objects are shown in the current example, this is for the purpose of illustration only and the process can be performed with any number of objects. For example, the environment may only include one or more movable objects, and may not include any other objects.

[0090] The system for controlling movement of objects typically includes one or more electronic processing devices 210, configured to receive image streams from imaging devices 220, which are provided in the environment *E* in order to allow images to be captured of the objects 160, 201, 202, 203 within the environment *E*.

[0091] For the purpose of illustration, it is assumed that the one or more electronic processing devices form part of one or more processing systems, such as computer systems, servers, or the like, which may be connected to one or more client devices, such as mobile phones, portable computers, tablets, or the like, via a network architecture, as will be described in more detail below. Furthermore, for ease of illustration the remaining description will refer to a processing device, but it will be appreciated that multiple processing devices could be used, with processing distributed between the processing devices as needed, and that reference to the singular encompasses the plural arrangement and *vice versa*.

[0092] The nature of the imaging devices 220 will vary depending upon the preferred implementation, but in one example the imaging devices are low cost imaging devices, such as non-computer vision monoscopic cameras. In one particular example, security cameras can be used, although as will become apparent from the following description, other low-cost cameras, such as webcams, or the like, could additionally and/or alternatively be used. It is also possible for a wide range of different imaging devices 220 to be used, and there is no need for the imaging devices 220 to be of a similar type or model.

[0093] The imaging devices 220 are typically statically positioned within the environment so as to provide coverage over a full extent of the environment *E*, with at least some of the cameras including at least partially overlapping fields of view, so that any objects within the environment *E* are preferably imaged by two or more of the imaging devices at any one time. The imaging may also be provided in a range of different positions in order to provide complete coverage. For example, the imaging devices could be provided at different heights, and could include a combination of floor, wall and/or ceiling mounted cameras, configured to capture views of the environment from different angles.

[0094] Operation of the system will now be described in more detail with reference to Figure 3.

[0095] In this example, at step 300, the processing device 210 receives image streams from each of the imaging devices 220, with the image streams including a plurality of captured images, and including images of the objects 160, 201, 202, 203, within the environment *E*.

[0096] At step 310 the one or more processing devices analyse the images to determine one or more object locations within the environment at step 320. In this regard, the object locations will include the location of any movable objects 160 and optionally, the location of any other objects 201, 202, 203, although this is not essential and may depending on the preferred implementation.

[0097] In one example, this analysis is performed by identifying overlapping images in the different image streams, which are images captured by imaging devices that have different overlapping fields of view, so that an image of an object is captured from at least two different directions by different imaging devices. In this case, the images are analysed to identify a position of an object in the different overlapping images, with knowledge of the imaging device locations being used to triangulate the position of the object. This can be achieved utilising any appropriate technique and in one example is achieved utilising a visual hull approach, as described for example in A. Laurentini (February 1994). "The visual hull concept for silhouette-based image understanding". IEEE Trans. Pattern Analysis and Machine Intelligence. pp. 150–162. However, other approaches, such image analysis of fiducial markings, can also be used. In one preferred example, fiducial markings are used in conjunction with the overlapping images, so that the object location can be determined with a higher degree of accuracy, although it will be appreciated that this is not essential.

[0098] At step 330, the one or more processing devices use the object location to generate control instructions, with the control instructions being provided to the controllers 154 of one or more modular wheels 150 at step 340, allowing these to be controlled to thereby move the object 160.

[0099] The manner in which the control instructions are generated, will vary depending on the preferred implementation. For example, each movable object may have associated routing information, such as an intended travel path, and/or an intended destination, in which case the control instructions can be generated to take into account the routing information and the current location. Additionally, and/or alternatively, control instructions can be generated to take into account locations of obstacles within the environment *E*, for example to ensure the object 160 does not collide with other objects 201, 202, 203, or parts of the environment *E*.

[0100] Control instructions could be generated to independently control each of the modular wheels 150 associated with an object 160, so that the controllers 154 are used to implement the respective control instructions provided. For example, the processing devices 210 could generate instructions specify a rate and/or amount of rotation for each modular wheel 150 associated with an object 160. However, alternatively, the processing devices 210 could simply identify a direction and/or rate of movement for the object as a whole, with the controllers 154 of each modular wheel 150 communicating to resolve how each wheel needs to be moved in order to result in the overall desired object movement.

[0101] In any event, it will be appreciated that the above described arrangement provides a system that can be used to move objects within an environment. Specifically, the system can be used to allow one or more modular wheels to be attached to an object, with a location of the object within the environment being monitored using a number of remote imaging devices positioned within the environment. Images from the imaging devices are analysed to monitor a location of the object within the environment, with this information being used to control movement of the object by controlling the one or more modular wheels.

[0102] Such an arrangement avoids the need for each object and/or each wheel to be fitted with sensors in order to navigate within the environment, whilst still allowing autonomous control of the object to be achieved. As sensing elements tend to be expensive and represent additional complexity, this helps ensure the modular wheel is cheap and relatively simple, allowing this to be more widely deployed than if inbuilt sensing were required.

[0103] In one example, this arrangement further allows a number of different movable objects to be controlled using common sensing apparatus, and allowing the movements to be centrally coordinated, which can in turn help optimise movement, and avoid potential collisions. For example, this allows modular wheels to be easily deployed within environments, such as factories, or the like. Specifically, in one example, this allows objects to be fitted with modular wheels and moved around as if the object were an AGV, thereby reducing the need for purpose built AGVs.

[0104] In one example, the object location detection and control can be performed by a system that is configured to perform situational awareness monitoring and an example of

such a system is described in copending patent application AU2019900442, the contents of which are incorporated herein by cross reference.

[0105] In this regard, situational awareness is the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their future status. Situational awareness is recognised as important for decision-making in a range of situations, particularly where there is interaction between people and equipment, which can lead to injury, or other adverse consequences. One example of this is within factories, where interaction between people and equipment has the potential for injury or death.

[0106] In one example, movement and/or locations of movable objects 160 and/or other objects 201, 202, 203 can be compared to situational awareness rules that define criteria representing desirable and/or potentially hazardous or other undesirable movement of movable objects 160 and/or other objects 201, 202, 203 within the environment *E*. The nature of the situational awareness rules and the criteria will vary depending upon factors, such as the preferred implementation, the circumstances in which the situational awareness monitoring process is employed, the nature of the objects being monitored, or the like. For example, the rules could relate to proximity of objects, a future predicted proximity of objects, interception of travel paths, certain objects entering or leaving particular areas, or the like, and examples of these will be described in more detail below.

[0107] Results of comparison of situational awareness rules can be used to identify any situational awareness events arising, which typically correspond to non-compliance or potential non-compliance with the situational awareness rules. Thus, for example, if a situational awareness rule is breached, this could indicate a potential hazard, which can, in turn, be used to allow action to be taken. The nature of the action and the manner in which this is performed will vary depending upon the preferred implementation.

[0108] For example, the action could include simply recording details of the situational awareness event, allowing this to be recorded for subsequent auditing purposes. Additionally, and/or alternatively, action can be taken in order to try and prevent hazardous situations, for example to alert individuals by generating a notification or an audible and/or visible alert.

This can be used to alert individuals within the environment so that corrective measures can be taken, for example by having an individual adjust their current movement or location, for example to move away from the path of an AGV. It will be appreciated that in the case of movable objects 160, as well as autonomous or semi-autonomous vehicles, this could include controlling the object/vehicle, for example by instructing the vehicle to change path or stop, thereby preventing accidents occurring.

[0109] Accordingly, it will be appreciated that the above described arrangement provides a system for moving objects within an environment using a modular wheel. The system can be configured to monitor the environment, allowing movement of movable objects to be controlled. Additionally, this can also be used to allow a location and/or movement of both movable and other objects, such as AGVs, people, to be tracked, which can in turn be used together with situational awareness rules to establish when undesirable conditions arise, such as when an AGV or other vehicle is likely to come into contact or approach a person, or *vice versa*. In one example, the system allows corrective actions to be performed automatically, for example to modify operation of the movable object, vehicle, or alert a person to the fact that a vehicle is approaching.

[0110] To achieve this, the system utilises a plurality of imaging devices provided within the environment *E* and operates to utilise images, typically with overlapping fields of view, in order to identify object locations. This avoids the need for objects to be provided with a detectable feature, such as RFID tags, or similar, although the use of such tags is not excluded as will be described in more detail below.

[0111] Furthermore, this process can be performed using low cost imaging devices, such as security cameras, or the like to be used, which offers a significant commercial value over other systems. The benefit is that the cameras are more readily available, known well and have a lower cost as well as can be connected to a single Ethernet cable that communicates and supplies power. This can then connect to a standard off the shelf POE Ethernet Switch as well as standard Power supply regulation. Thus, this can vastly reduce the cost of installing and configuring such a system, avoiding the need to utilise expensive stereoscopic or

computer vision cameras and in many cases allows existing security camera infrastructure to be utilised for situational awareness monitoring.

[0112] Additionally, the processing device can be configured to perform the object tracking and control, as well as the identification of situational awareness events, or actions, substantially in real time. Thus, for example, the time taken between image capture and an action being performed can be less than about 1 second, less than about 500ms, less than about 200ms, or less than about 100ms. This enables the system to effectively control movable objects and/or take other corrective action, such as alerting individuals within the environment, controlling vehicles, or the like, thereby allowing events, such as impacts or injuries to be avoided.

[0113] A number of further features will now be described.

[0114] As mentioned above, in one example, the system includes one or more passive wheels mounted to the object. Such passive wheels could be multi-directional wheels, such as castor wheels, or similar, in which case the controller(s) can be configured to steer the object through differential rotation of two or more modular wheels. Additionally, and/or alternatively, the modular wheel can include a steering drive configured to adjust an orientation of the wheel, in which case the controller(s) can be configured to control the steering drive to thereby change an orientation of the wheel, and hence direct movement of the movable object. It will also be appreciated that other configurations could be used, such as providing drive wheels and separate steering wheels. However, in general, providing both steering and drive in single modular wheels provides a greater range in flexibility, allowing identical modular wheels to be used in a range of different ways. This can also assist in addressing wheel failure, for example allowing differently control modes to be used if one or more of the modular wheels fail.

[0115] In one example, the modular wheel includes a transceiver configured to communicate wirelessly with the one or more processing devices. This allows each modular wheel to communicate directly with the processing devices, although it will be appreciated that this is not essential, and other arrangements, such as using a centralised communications module, mesh networking between multiple modular wheels, or the like, could be used.

[0116] Each modular wheel typically includes a power supply, such as battery, configured to power the drive, the controller, the transceiver, steering drive, and any other components. Providing a battery for each wheel, allows each wheel to be self-contained, meaning the wheel need only be fitted to the object, and does not need to be separately connected to a power supply or other wheel, although it will be appreciated that separate power supplies could be used depending on the intended usage scenario.

[0117] In one example, the system includes a plurality of modular wheels and the processing device is configured to provide respective control instructions to each controller to thereby independently control each modular wheel. For example, this could include having the processing device generate control instructions including a wheel orientation and/or a rate of rotation for each individual modular wheel.

[0118] In another example, the processing devices are configured to provide control instructions to the controllers and wherein the controllers of different modular wheels communicate to independently control each modular wheel. For example, the processing devices could generate control instructions including a direction and rate of travel for the object, with the controller for each modular wheel attached to that object then collaboratively determining a wheel orientation and/or rate of rotation for each wheel. In a further example, a master slave arrangement could be used, allowing a master modular wheel to calculate movements for each individual modular wheel, with that information being communicated to the other modular wheel controllers as needed.

[0119] In one example, the processing device is configured to determine an object configuration and then generate the control instructions at least partially in accordance with the object configuration. The object configuration could be indicative of anything that can influence movement of the object, and could include an object extent, parameters relating to movement of the object or the like. This allows the processing device to take into a size and/or shape of the object, when controlling the wheels, thereby ensuring the object does not impact on other objects or parts of the environment. This is particularly important for irregular sized objects and/or objects with overhangs, which might otherwise collide with

obstacles such as the environment and/or other objects, even when the wheels are distant from the obstacle.

[0120] In this regard, it will be appreciated that as the object location is detected using image analysis, this same approach can be used to determine the object extent. For example, a visual hull analysis can be used, examining images of the object from multiple viewpoints, in order to calculate the size and/or shape of the object.

[0121] Similarly, the processing devices can be configured to determine a wheel configuration indicative of a position of each wheel relative to the object and/or a relative position of each wheel, and generate the control instructions at least partially in accordance with the wheel configuration. Thus, in a similar way, the processing devices can use image analysis to detect the position of the wheels relative to the object and then use this information when generating the control signals. This can be used to ensure that the wheels operate collectively so that movement of the object is accurately controlled, although it will be appreciated that alternatively the one or more modular wheels can be at known locations.

[0122] Alternatively, an object and/or wheel configurations can be predefined, for example during an initial set-up process, and then retrieved as needed.

[0123] In one example, the processing device is configured to determine an identity of one or more modular wheels or the object and then generate control instructions in accordance with the identity. For example, this can be used to ensure that control instructions are transmitted to the correct modular wheel. This could also be used to allow the processing device to retrieve an object or wheel configuration, allowing such configurations to be stored and retrieved based on the object and/or wheel identity as needed.

[0124] The identity could be determined in any one of a number of manners depending on the preferred implementation. For example, the wheels could be configured to communicate with the processing device via a communication network, in which case the processing device could be configured to determine the identity at least in part using a network identifier, such as an IP (Internet Protocol) or MAC (Media Access Control) address, or similar. In another example, the processing devices can be configured to determine the

identity using machine readable coded data. This could include visible coded data provided on the object and/or wheels, such as a bar code, QR code or more typically an April Tag, which can then be detected by analysing images to identify the visible machine readable coded data in the image allowing this to be decoded by the processing device. In another example however objects and/or modular wheels may be associated with tags, such as short range wireless communication protocol tags, RFID (Radio Frequency Identification) tags, Bluetooth tags, or similar, in which case the machine readable coded data could be retrieved from a suitable tag reader.

[0125] It will also be appreciated that different identification approaches could be used in conjunction. For example, an object and/or wheel could be uniquely identified through detection of machine-readable coded data when passing by a suitable reader. In this instance, once the object has been identified, this identity can be maintained by keeping track of the object as it moves within the environment. This means that the object does not need to pass by a reader again in order to be identified, which is particularly useful in circumstances where objects are only identified on limited occasions, such as upon entry into an area.

[0126] Typically the one or more processing devices are configured to determine routing data indicative of a travel path and/or a destination, and then generate control instructions in accordance with the routing data and the object location. For example, routing data could be retrieved from a data store, such as a database, using an object and/or wheel identity.

[0127] In addition to indicating a travel path and/or destination, the routing data could also be indicative of a permitted object travel path, permitted object movements, permitted proximity limits for different objects, permitted zones for objects or denied zones for objects. This additional information could be utilised in the event a preferred path cannot be followed, allowing alternative routes to be calculated, for example to avoid obstacles, such as other objects.

[0128] Similarly, the processing device can be configured to identify if an object movement deviates from a permitted object travel path defined for the object, an object movement deviates from a permitted object movement for the object, two objects are within permitted proximity limits for the objects, two objects are approaching permitted proximity limits for

the objects, two objects are predicted to be within permitted proximity limits for the objects, two objects have intersecting predicted travel paths, an object is outside a permitted zone for the object, an object is exiting a permitted zone for the object, an object is inside a denied zone for the object or an object is entering a denied zone for the object. This can then be used to take corrective measures, including stopping or correcting movement.

[0129] In one example, particularly in the event that a wheel configuration is unknown, the processing device could be configured to generate calibration control instructions, which are used to induce minor movements of one or more wheels, such as performing a single or part revolution, or a defined change in wheel orientation. The processing device can then be configured to monitor movement of the object and/or the modular wheel in response to the calibration control instructions and use results of the monitoring to generate control instructions.

[0130] For example, by generating calibration instructions and then monitoring movement, this can be used to identify particular wheels, for example matching an IP address of a wheel with a particular wheel on a specific object. This can also be used to identify how the wheel responds, which in turn can be used to accurately control the wheel, for example to account for a mounting orientation of the wheel on the object. Accordingly, this allows the system to use visual feedback in order to calculate a wheel configuration, hence allowing control instructions to be generated that can be used to accurately control a wheel.

[0131] In one example, the imaging devices are positioned within the environment at fixed locations and are static, or at least substantially static, relative to the environment. This allows the images captured by each camera or other imaging device to be more easily interpreted, although it will be appreciated that this is not essential and alternatively moving cameras, such as cameras that repeatedly pan through the environment, could be used. In this case the processing device is typically configured to analyse the images taking into account a position and/or orientation of the wheels.

[0132] Typically, the imaging devices are positioned within the environment to have at least partially overlapping fields of view and wherein the one or more processing devices are configured to identify overlapping images in the different image streams, the overlapping

images being images captured by imaging devices having overlapping fields of view and then analyse the overlapping images to determine object locations within the environment.

[0133] In one example, the overlapping images are synchronous overlapping images in that they are captured at approximately the same time. In this regard, the requirement for images to be captured at approximately the same time means that the images are captured within a time interval less than that which would result in substantial movement of the object. Whilst this will therefore be dependent on the speed of movement of the objects, the time interval is typically less than about 1 second, less than about 500ms, less than about 200ms, or less than about 100ms.

[0134] In order to identify synchronous overlapping images, the processing device is typically configured to synchronise the image streams, typically using information such as a timestamp associated with the images and/or a time of receipt of the images from the imaging device. This allows images to be received from imaging devices other than computer vision devices, which are typically time synchronised with the processing device. Thus, this requires additional functionality to be implemented by the processing device in order to ensure accurate time sync between all of the camera feeds which is normally performed in hardware.

[0135] In one example, the processing devices are configured to determine a captured time of each captured image, and then identify the synchronous images using the captured time. This is typically required because the imaging devices do not incorporate synchronisation capabilities, as present in most computer vision cameras, allowing the system to be implemented using cheaper underlying and existing technologies, such as security cameras, or the like.

[0136] The manner in which a capture time is determined for each captured image will vary depending upon the preferred implementation. For example, the imaging device may generate a captured time, such as a time stamp, as is the case with many security cameras. In this case, by default the time stamp can be used as the capture time. Additionally and/or alternatively, the captured time could be based on a receipt time indicative of a time of receipt by the processing device. This might optionally take into account a communication

delay between the imaging device and the processing device, which could be established during a calibration, or other set up process. In one preferred example, the two techniques are used in conjunction, so that the time of receipt of the images is used to validate a time stamp associated with the images, thereby providing an additional level of verification to the determined capture time, and allowing corrective measures to be taken in the event a capture time is not verified.

[0137] However, it will be appreciated that the use of synchronous images is not essential, and asynchronous images or other data could be used, depending on the preferred implementation. For example, images of an object that are captured asynchronously can result in the object location changing between images. However, this can be accounted for through suitable techniques, such as weighting images, so images captured asynchronously are given a temporal weighting in identifying the object location, as will be described in more detail below. Other techniques could also be employed such as assigning an object a fuzzy boundary and/or location, or the like.

[0138] In one example, the processing devices are configured to analyse images from each image stream to identify object images, which are images including objects. Having identified object images, these can be analysed to identify overlapping images as object images that include an image of the same object. This can be performed on the basis of image recognition processes but more typically is performed based on knowledge regarding the relative position, and in particular fields of view, of the imaging devices. This may also take into account for example a position of an object within an image, which could be used to narrow down overlapping fields of view between different imaging devices to thereby locate the overlapping images. Such information can be established during a calibration process as will be described in more detail below.

[0139] In one example, the one or more processing devices are configured to analyse a number of images from an image stream to identify static image regions and then identify object images as images including non-static image regions. In particular, this involves comparing successive or subsequent images to identify movement that occurs between the images, with it being assessed that the movement component in the image is as a result of an

object moving. This relies on the fact that the large majority of the environment will remain static, and that in general it is only the objects that will undergo movement between the images. Accordingly, this provides an easy mechanism to identify objects, and reduces the amount of time required to analyse images to detect objects therein.

[0140] It will be appreciated that in the event that an object is static, it may not be detected by this technique if the comparison is performed between successive images. However, this can be addressed by applying different learning rates to background and foreground. For example, background reference images could be established in which no objects are present in the environment, with the subtraction being performed relative to the background reference images, as opposed to immediately preceding images. In one preferred approach, the background reference images are periodically updated to take into account changes in the environment. Accordingly, this approach allows stationary objects to be identified or tracked. It will also be appreciated that tracking of static objects can be performed utilising an environment model that maintains a record of the location of static objects so that tracking can resume when the object recommences movement, as will be described in more detail below.

[0141] Thus, the one or more processing devices can be configured to determine a degree of change in appearance for an image region between images in an image stream and then identify objects based on the degree of change. The images can be successive images and/or temporally spaced images, with the degree of change being a magnitude and/or rate of change. The size and shape of the image region can vary, and may include subsets of pixels, or similar, depending on the preferred implementation. In any event, it will be appreciated that if the image region is largely static, then it is less likely that an object is within the image region, than if the image region is undergoing significant change in appearance, which is indicative of a moving object.

[0142] In one particular example, this is achieved by classifying image regions as static or non-static image regions (also referred to as background and foreground image regions), with the non-static image regions being indicative of movement, and hence objects, within the environment. The classification can be performed in any appropriate manner, but in one

example, this is achieved by comparing a degree of change to a classification threshold and then classifying the image region based on results of the comparison, for example classifying image regions as non-static if the degree of change exceeds the classification threshold, or static if the degree of change is below the classification threshold.

[0143] Following this, objects can be identified based on classification of the image region, for example by analysing non-static image regions to identify objects. In one example, this is achieved by using the image regions to establish masks, which are then used in performing subsequent analysis, for example, with foreground masks being used to identify and track objects, whilst background masks are excluded to reduce processing requirements.

[0144] Additionally, in order to improve discrimination, the processing device can be configured to dynamically adjust the classification threshold, for example to take into account changes in object movement, environmental effects or the like.

[0145] For example, if an object ceases moving, this can result in the image region being reclassified as static, even though it contains an object. Accordingly, in one example, this can be accounted for by having the processing device identify an object image region containing an object that has previously been moving, and then modifying the classification threshold for the object image region so as to decrease the degree of change required in order to classify the image region as a non-static image region. This in effect changes the learning rate for the region as mentioned above. As changes in the appearance of the image region can be assessed cumulatively, this in effect can increase the duration from when movement within a region stops to the time at which the region is classified as a static image region, and hence is assessed as not containing an object. As a result, if an object stops moving for a relatively short period of time, this avoids the region being reclassified, which can allow objects to be tracked more accurately.

[0146] In another example, the processing devices can be configured to identify images including visual effects and then process the images in accordance with the visual effects to thereby identify objects. In this regard, visual effects can result in a change of appearance between images of the same scene, which could therefore be identified as a potential object. For example, an AGV may include illumination ahead of the vehicle, which could be

erroneously detected as an object separate from the AGV as it results in a change in appearance that moves over time. Similar issues arise with other changes in ambient illumination, such as changes in sunlight within a room, the presence of visual presentation devices, such as displays or monitors, or the like.

[0147] In one example, to address visual effects, the processing device can be configured to identify image regions including visual effects and then exclude image regions including visual effects and/or classify image regions accounting for the visual effects. Thus, this could be used to adjust a classification threshold based on identified visual effects, for example by raising the classification threshold so that an image region is less likely to be classified as a non-static image region when changes in visual appearance are detected.

[0148] The detection of visual effects could be achieved using a variety of techniques, depending on the preferred implementation, available sensors and the nature of the visual effect. For example, if the visual effect is a change in illumination, signals from one or more illumination sensors could be used to detect the visual effect. Alternatively, one or more reference images could be used to identify visual effects that routinely occur within the environment, such as to analyse how background lighting changes during the course of a day, allowing this to be taken into account. This could also be used in conjunction with environmental information, such as weather reports, information regarding the current time of day, or the like, to predict likely illumination within the environment. As a further alternative, manual identification could be performed, for example by having a user specify parts of the environment that could be subject to changes in illumination and/or that contain monitors or displays, or the like.

[0149] In another example, visual effects could be identified by analyzing images in accordance with defined properties thereby allowing regions meeting those properties to be excluded. For example, this could include identifying illumination having known wavelengths and/or spectral properties, such as corresponding to vehicle warning lights, and then excluding such regions from analysis.

[0150] In general, the processing device is configured to analyse the synchronous overlapping images in order to determine object locations. In one example, this is performed

using a visual hull technique, which is a shape-from-silhouette 3D reconstruction technique. In particular, such visual hull techniques involve identifying silhouettes of objects within the images, and using these to create a back-projected generalized cone (known as a "silhouette cone") that contains the actual object. Silhouette cones from images taken from different viewpoints are used to determine an intersection of the two or more cones, which forms the visual hull, which is a bounding geometry of the actual 3D object. This can then be used to ascertain the location based on known viewpoints of the imaging devices. Thus, comparing images captured from different viewpoints allows the position of the object within the environment E to be determined.

[0151] However, in some examples, such a visual hull approach is not required. For example, if the object includes machine readable visual coded data, such as a fiducial marker, or an April Tag, described in "AprilTag: A robust and flexible visual fiducial system" by Edwin Olson in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011, then the location can be derived through a visual analysis of the fiducial markers in the captured images.

[0152] In one preferred example, the approach uses a combination of fiducial markings and multiple images, allowing triangulation of different images containing the fiducial markings, to allowing the location of the fiducial markings and hence the object, to be calculated with a higher degree of accuracy.

[0153] In one example, particularly when using visual hull or similar techniques, the processing device can be configured to identify corresponding image regions in overlapping images, which are images of a volume within the environment, and then analyse the corresponding image regions to identify objects in the volume. Specifically, this typically includes identifying corresponding image regions that are non-static image regions of the same volume that have been captured from different points of view. The non-static regions can then be analysed to identify candidate objects, for example using the visual hull technique.

[0154] Whilst this process can be relatively straightforward when there is a sufficiently high camera density, and/or objects are sparsely arranged within the environment, this process

becomes more complex when cameras are sparse and/or objects densely arranged and/or are close to each other. In particular, in this situation, objects are often wholly or partially occluded, meaning the shape of the object may not be accurately captured using some of the imaging devices. This can in turn lead to misidentification of objects, or their size, shape or location.

[0155] Accordingly, in one example, once candidate objects have been identified, these can be incorporated into a three dimensional model of the environment, and examples of such a model are described in more detail below. The model can then be analysed to identify potential occlusions, for example when another object is positioned between the object and an imaging device, by back-projecting objects from the three dimensional model into the 2D imaging plane of the imaging device. Once identified, the potential occlusions can then be used to validate candidate objects, for example allowing the visual hull process to be performed taking the occlusion into account. Examples of these issues are described in "Visual Hull Construction in the Presence of Partial Occlusion" by Li Guan, Sudipta Sinha, Jean-Sebastien Franco, Marc Pollefeys.

[0156] In one example, the potential occlusions are accounted for by having the processing device classify image regions as occluded image regions if they contain potential occlusions, taking this into account when identifying the object. Whilst this could involve simply excluding occluded image regions from the analysis, typically meaningful information would be lost with this approach, and as a result objects may be inaccurately identified and/or located. Accordingly, in another example, this can be achieved by weighting different ones of the corresponding images, with the weighting being used to assess the likely impact of the occlusion and hence the accuracy of any resulting object detection.

[0157] A similar weighting process can also be used to take other issues into account. Accordingly, in one example, an object score is calculated associated with corresponding image regions, with the object score being indicative of certainty associated with detection of an object in the corresponding image regions. Once calculated the object score can be used to identify objects, for example assessing an object to be accurately identified if the score exceeds a score threshold.

[0158] It will be appreciated that the object score could be used in a variety of manners. For example, if an object score is too low, an object could simply be excluded from the analysis. More usefully however, this could be used to place constraints on how well known is the object location, so that a low score could result in the object having a high degree of uncertainty on the location, allowing this to be taken into account when assessing situational awareness.

[0159] In one example, the object score is calculated by generating an image region score for each of the corresponding image regions and then calculating the object score using the image region scores for the image regions of each of the corresponding images. Thus image region scores are calculated for each imaging device that captures an image region of a particular volume, with these being combined to create an overall score for the volume.

[0160] Additionally and/or alternatively, the individual image region scores could be used to assess the potential reliability for an image region to be used in accurately identifying an object. Thus if the likely accuracy of any particular image region is low, such as if there is a significant occlusion, or the like, this could be given a low weighting in any analysis, so it is less likely to adversely impact on the subsequent analysis, thereby reducing the chance of this unduly influencing the resulting object location.

[0161] As mentioned, whilst this approach could be used for occlusions, this could also be used for a wide range of other factors. For example, the score could be based on an image region classification, or a degree of change in appearance for an image region between images in an image stream, so that static image regions that are unlikely to contain an object have a low score, whereas non-static regions could have a higher score.

[0162] Similarly, this approach could take into account longevity of an image region classification or historical changes image region classification. Thus, if the classification of an image region changes frequently, this might suggest the region is not being correctly classified, and hence this could be given a low confidence score, whereas a constant classification implies a higher confidence in correct classification and hence a higher score.

[0163] Similarly, a score could be assigned based on a presence or likelihood of visual effects, allowing visual effects to be accounted for when identifying objects. In another example, a camera geometry relative to the volume of interest could be used, so that images captured by a camera that is distant from, or obliquely arranged relative to, a volume are given a lower weighting.

[0164] The factors could also take into account a time of image capture, which can assist in allowing asynchronous data capture to be used. In this instance, if one of the overlapping images is captured at a significantly different time to the other images, this may be given a lower weighting in terms of identifying the object given the object may have moved in the intervening time.

[0165] Similarly, other factors relating to image quality, such as resolution, focus, exposure, or the like, could also be taken into account.

[0166] Accordingly, it will be appreciated that calculating a score for the image captured by each imaging device could be used to weight each of the images, so that the degree to which the image is relied upon in the overall object detection process can take into account factors such as the quality of the image, occlusions, visual effects, how well the object is imaged, or the like, which can in turn allow object detection to be performed more accurately.

[0167] Irrespective of how objects are detected, determining the location of the objects typically requires knowledge of the positioning of the imaging devices within the environment, and so the processing device is configured to interpret the images in accordance with a known position of the image devices. To achieve this, in one example, position information is embodied in calibration data which is used in order to interpret the images. In one particular example, the calibration data includes intrinsic calibration data indicative of imaging properties of each imaging device and extrinsic calibration data indicative of the relative positioning of the imaging devices within the environment *E*. This allows the processing device to correct images to account for any imaging distortion in the captured images, and also to account for the position of the imaging devices relative to the environment *E*.

[0168] The calibration data is typically generated during a calibration process. For example, intrinsic calibration data can be generated based on images of defined patterns captured from different positions using an imaging device. The defined patterns could be of any appropriate form and may include patterns of dots or similar, fiducial markers, or the like. The images are analysed to identify distortions of the defined patterns in the image, which can in turn be used to generate calibration data indicative of image capture properties of the imaging device. Such image capture properties can include things such as a depth of field, a lens aperture, lens distortion, or the like.

[0169] In contrast, extrinsic calibration data can be generated by receiving captured images of targets within the environment, analysing the captured images to identify images captured by different imaging devices which show the same target and analysing the identified images to generate calibration data indicative of a relative position and orientation of the imaging devices. Again this process can be performed by positioning multiple targets within the environment and then identifying which targets have been captured using which imaging devices. This can be performed manually, based on user inputs, or can be performed automatically using unique targets so that different targets and different positions can be easily identified. It will be appreciated from this that fiducial markers, such as April Tags could be used for this purpose.

[0170] As mentioned above, in one example, the system can be part of a situational awareness monitoring system. In this example, in addition to controlling movement using the modular wheels as described above, the system uses situational awareness rules to identify situational awareness events, in turn allowing an action to be performed, such as attempting to mitigate the event.

[0171] The nature of the situational awareness rules will vary depending on the preferred implementation, as well as the nature of the environment, and the objects within the environment. In one example, the situational awareness rules are indicative of one or more of permitted object travel paths, permitted object movements, permitted proximity limits for different objects, permitted zones for objects, or denied zones of objects. In this example, situational awareness events could be determined to occur if an object movement deviates

from a permitted object travel path, if object movement deviates from a permitted object movement, if two objects are within a predetermined proximity limit for the objects, if two objects are approaching permitted proximity limits for the objects, if two objects have intersecting predicted travel paths, if an object is outside a permitted zone for the object, if an object is exiting a permitted zone for the object, if an object is inside a denied zone for the object, if an object is entering a denied zone for the object, or the like. It will be appreciated however that a wide range of different situational awareness rules could be defined to identify a wide range of different situational awareness events, and that the above examples are for the purpose of illustration only.

[0172] Such rules can be generated using a variety of techniques, but are typically generated manually through an understanding of the operation and interaction of objects in the environment. In another example, a rules engine can be used to at least partially automate the task. The rules engine typically operates by receiving a rules document, and parsing using natural language processing, to identify logic expressions and object types. An object identifier is then determined for each object type, either by retrieving these based on an object type of the object or by generating these as needed. The logic expressions are then used to generate the object rules by converting the logic expressions into a trigger event and an action, before uploading these to the tag. For example, the logic expressions are often specified within a rules text in terms of "If... then..." statements, which can be converted to a trigger and action. This can be performed using templates, for example by populating a template using text from the "If... then..." statements, so that the rules are generated in a standard manner, allowing these to be interpreted consistently.

[0173] Once rules are generated, actions to be taken in response to a breach of the rules can also be defined, with this information being stored as rules data in a rules database.

[0174] It will be appreciated from the above that different rules are typically defined for different objects and/or object types. Thus, for example, one set of rules could be defined for individuals, whilst another set of rules might be defined for AGVs. Similarly, different rules could be defined for different AGVs which are operating in a different manner. It will also

be appreciated that in some instances rules will relate to interaction between objects, in which case the rules may depend on the object identity of multiple objects.

[0175] Accordingly, in one example, when the rules are created, the rules are associated with one or more object identities, which can be indicative of an object type and/or can be uniquely indicative of the particular object, allowing object situational awareness rules to be defined for different types of objects, or different individual objects. This allows the processing device to subsequently retrieve relevant rules based on the object identities of objects within the environment. Accordingly, in one example, the one or more processing devices are configured to determine an object identity for at least one object and then compare the object movement to situational awareness rules at least in part using the object identity. Thus, the processing device can select one or more situational awareness rules in accordance with the object identity, and then compare the object movement to the selected situational awareness rule.

[0176] The object identity can be determined utilising a number of different techniques, depending for example on the preferred implementation, and/or, the nature of the object. For example, if the object does not include any form of encoded identifier, this could be performed using image recognition techniques. In the case of identifying people for example, the people will have a broadly similar appearance will generally be quite different to that of an AGV and accordingly, people could be identified using image recognition techniques performed on or more of the images. It will be noted that this does not necessarily require discrimination of different individuals, although this may be performed in some cases.

[0177] Additionally and/or alternatively, identities, and in particular object types could be identified through an analysis of movement. For example movement of AGVs will typically tend to follow predetermined patterns and/or have typically characteristics such as constant speed and/or direction changes. In contrast to this movement of individuals will tend to be more haphazard and subject to changes in direction and/or speed allowing, AGVs and humans to be distinguished based on an analysis of movement patterns.

[0178] In another example, an object can be associated with machine readable code data indicative of an object identity. In this example, the processing devices can be configured to determine the object identity using the machine readable coded data. The machine readable coded data could be encoded in any one of a number of ways depending on the preferred implementation. In one example, this can be achieved using visual coded data such as a bar code, QR code or more typically an April Tag, which can then be detected by analysing images to identify the visible machine readable coded data in the image allowing this to be decoded by the processing device. In another example however objects may be associated with tags, such as short range wireless communication protocol tags, RFID (Radio Frequency Identification) tags, Bluetooth tags, or similar, in which case the machine readable coded data could be retrieved from a suitable tag reader.

[0179] It will also be appreciated that identification approaches could be used in conjunction. For example, an object could be uniquely identified through detection of machine readable coded data when passing by a suitable reader. In this instance, once the object has been identified, this identity can be maintained by keeping track of the object as it moves within the environment. This means that the object does not need to pass by a reader again in order to be identified, which is particularly useful in circumstances where objects are only identified on limited occasions, such as upon entry into an area, as may occur for example when an individual uses an access card to enter a room.

[0180] In one example, the one or more processing devices are configured to use object movements to determine predicted object movements. This can be performed, for example, by extrapolating historical movement patterns forward in time, assuming for example that a object moving in a straight line will continue to move in a straight line for at least a short time period. It will be appreciated that a variety of techniques can be used to perform such predictions, such as using machine learning techniques to analyse patterns of movement for specific objects, or similar types of objects, combining this with other available information, such as defined intended travel paths, or the like, in order to predict future object movement. The predicted object movements can then be compared to situational awareness rules in order to identify potential situational awareness events in advance. This could be utilised, for example, to ascertain if an AGV and person are expected to intercept at some point in the

future, thereby allowing an alert or warning to be generated prior to any interception occurring.

[0181] In one example, the one or more processing devices are configured to generate an environment model indicative of the environment, a location of imaging devices in the environment, locations of client devices, such as alerting beacons, current object locations, object movements, predicted object locations, predicted object movements, or the like. The environment model can be used to maintain a record of recent historical movements, which in turn can be used to assist in tracking objects that are temporarily stationary, and also to more accurately identify situational awareness events.

[0182] The environment model may be retained in memory and could be accessed by the processing device and/or other processing devices, such as remote computer systems, as required. In another example, the processing devices can be configured to generate a graphical representation of the environment model, either as it currently stands, or at historical points in time. This can be used to allow operators or users to view a current or historical environment status and thereby ascertain issues associated with situational awareness events, such as reviewing a set of circumstances leading to an event occurring, or the like. This could include displaying heat maps, showing the movement of objects within the environment, which can in turn be used to highlight bottle necks, or other issues that might give rise to situational awareness events.

[0183] As previously mentioned, in response to identifying situational awareness events, actions can be taken. This can include, but is not limited to, recording an indication of the situational awareness event, generating a notification indicative of the situational awareness event, causing an output device to generate an output indicative of the situational awareness event, including generating an audible and/or visual output, activating an alarm, or causing operation of an object to be controlled. Thus, notifications could be provided to overseeing supervisors or operators, alerts could be generated within the environment, for example to notify humans of a potential situational awareness event, such as a likelihood of imminent collision, or could be used to control autonomous or semi-autonomous vehicles, such as

AGVs, allowing the vehicle control system to stop the vehicle and/or change operation of the vehicle in some other manner, thereby allowing an accident or other event to be avoided.

[0184] The imaging devices can be selected from any one or more of security imaging devices, monoscopic imaging devices, non-computer vision-based imaging devices or imaging devices that do not have intrinsic calibration information. Furthermore, as the approach does not rely on the configuration of the imaging device, handling different imaging devices through the use of calibration data, this allows different types and/or models of imaging device to be used within a single system, thereby providing greater flexibility to the equipment that can be used to implement the situational awareness monitoring system.

[0185] As mentioned above, in one example, the process is performed by one or more processing systems and optionally one or more client devices operating as part of a distributed architecture, an example of which will now be described with reference to Figure 4.

[0186] In this example, a number of processing systems 410 are coupled via communications networks 440, such as the Internet, and/or one or more local area networks (LANs), to a number of client devices 430 and imaging devices 420, as well as to a number of modular wheel controllers 454. It will be appreciated that the configuration of the networks 440 are for the purpose of example only, and in practice the processing systems 410, imaging devices 420, client devices 430 and controllers 454 can communicate via any appropriate mechanism, such as via wired or wireless connections, including, but not limited to mobile networks, private networks, such as an 802.11 networks, the Internet, LANs, WANs, or the like, as well as via direct or point-to-point connections, such as Bluetooth, or the like.

[0187] In one example, the processing systems 410 are configured to receiving image streams from the imaging devices 420, analyse the image streams, generate control signals for controlling the modular wheels, and optionally identify situational awareness events. The processing systems 410 can also be configured to implement actions, such as generating notifications and/or alerts, optionally displayed via client devices or other hardware, control operations of AGVs, or similar, or create and provide access to an environment model. Whilst the processing system 410 is shown as a single entity, it will be appreciated that the

- 38 -

processing system 410 can be distributed over a number of geographically separate locations, for example by using processing systems 410 and/or databases that are provided as part of a cloud based environment. However, the above described arrangement is not essential and other suitable configurations could be used.

[0188] An example of a suitable processing system 410 is shown in Figure 5.

[0189] In this example, the processing system 410 includes at least one microprocessor 511, a memory 512, an optional input/output device 513, such as a keyboard and/or display, and an external interface 514, interconnected via a bus 515, as shown. In this example the external interface 514 can be utilised for connecting the processing system 410 to peripheral devices, such as the communications network 440, databases, other storage devices, or the like. Although a single external interface 514 is shown, this is for the purpose of example only, and in practice multiple interfaces using various methods (eg. Ethernet, serial, USB, wireless or the like) may be provided.

[0190] In use, the microprocessor 511 executes instructions in the form of applications software stored in the memory 512 to allow the required processes to be performed. The applications software may include one or more software modules, and may be executed in a suitable execution environment, such as an operating system environment, or the like.

[0191] Accordingly, it will be appreciated that the processing system 410 may be formed from any suitable processing system, such as a suitably programmed client device, PC, web server, network server, or the like. In one particular example, the processing system 310 is a standard processing system such as an Intel Architecture based processing system, which executes software applications stored on non-volatile (e.g., hard disk) storage, although this is not essential. However, it will also be understood that the processing system could be any electronic processing device such as a microprocessor, microchip processor, logic gate configuration, firmware optionally associated with implementing logic such as an FPGA (Field Programmable Gate Array), or any other electronic device, system or arrangement.

[0192] An example of a suitable client device 430 is shown in Figure 6.

[0193] In one example, the client device 430 includes at least one microprocessor 631, a memory 632, an input/output device 633, such as a keyboard and/or display, and an external interface 634, interconnected via a bus 635, as shown. In this example the external interface 634 can be utilised for connecting the client device 430 to peripheral devices, such as the communications networks 440, databases, other storage devices, or the like. Although a single external interface 634 is shown, this is for the purpose of example only, and in practice multiple interfaces using various methods (eg. Ethernet, serial, USB, wireless or the like) may be provided.

[0194] In use, the microprocessor 631 executes instructions in the form of applications software stored in the memory 632 to allow communication with the processing system 410, for example to allow notifications or the like to be received and/or to provide access to an environment model.

[0195] Accordingly, it will be appreciated that the client devices 430 may be formed from any suitable processing system, such as a suitably programmed PC, Internet terminal, lap-top, or hand-held PC, and in one preferred example is either a tablet, or smart phone, or the like. Thus, in one example, the client device 430 is a standard processing system such as an Intel Architecture based processing system, which executes software applications stored on non-volatile (e.g., hard disk) storage, although this is not essential. However, it will also be understood that the client devices 430 can be any electronic processing device such as a microprocessor, microchip processor, logic gate configuration, firmware optionally associated with implementing logic such as an FPGA (Field Programmable Gate Array), or any other electronic device, system or arrangement.

[0196] An example of a modular wheel will now be described in more detail with reference to Figures 7A and 7B.

[0197] In this example, the modular wheel 750 includes a body 751 having a mounting 757 configured to be attached to the object. The body has a “7” shape, with an upper lateral portion 751.1 supporting the mounting 757, and an inwardly sloping diagonal leg 751.2 extending down to a hub 751.3 that supports the wheel 752. A drive 753 is attached to the hub, allowing the wheel to be rotated. A battery 756 is mounted on an underside of the

sloping diagonal leg 751.2, with a controller 754 being mounted on an outer face of the battery. A steering drive 755 is also provided, which allows the body 751 to be rotated relative to the mounting 757, thereby allowing an orientation (heading) of the wheel to be adjusted.

[0198] In one specific example, the modular wheel is designed to be self-contained two degree of freedom wheels. Each modular wheel can generate a speed and heading through the use of continuous rotation servos located behind the wheel and below the coupling at the top of the module. Their centres of rotation aligning to reduce torque during rotation. The wheel and top coupling use ISO 9409-1404M6 bolt pattern to enable cross platform comparability. A generic set of adaptors can be used to enable rapid system assembly and reconfiguration.

[0199] An example of a suitable controller is shown in Figure 8.

[0200] In one example, the controller 754 includes at least one microprocessor 871, a memory 872, an input/output device 873, such as a keyboard and/or display, and an external interface 874, interconnected via a bus 875, as shown. In this example the external interface 874 can be utilised for connecting the controller 754 to peripheral devices, such as the communications networks 440, databases, other storage devices, or the like. Although a single external interface 874 is shown, this is for the purpose of example only, and in practice multiple interfaces using various methods (eg. Ethernet, serial, USB, wireless or the like) may be provided.

[0201] In use, the microprocessor 871 executes instructions in the form of applications software stored in the memory 872 to allow communication with the processing system 410, for example to allow notifications or the like to be received and/or to provide access to an environment model.

[0202] Accordingly, it will be appreciated that the controllers 754 may be formed from any suitable processing system, such as a suitably programmed PC, Internet terminal, lap-top, or hand-held PC, and in one preferred example is either a tablet, or smart phone, or the like. Thus, in one example, the controller 754 is a standard processing system, which executes software applications stored on non-volatile (e.g., hard disk) storage, although this is not

essential. However, it will also be understood that the controller 754 can be any electronic processing device such as a microprocessor, microchip processor, logic gate configuration, firmware optionally associated with implementing logic such as an FPGA (Field Programmable Gate Array), or any other electronic device, system or arrangement.

[0203] In one specific example, the controller 754 is in the form of a Raspberry Pi providing both the wheels commands and wi-fi communication between modular wheels and/or communications networks. Built into the body or leg of each wheel is a four cell lithium polymer battery providing power. The battery can be accessed through a removable panel.

[0204] In one example, central control of the modular wheel system uses relative velocities to set the velocity, and hence rotation rate, of the individual modular wheels. Each modular wheels pose (position and orientation) relative to the centre of the object can be used to determine the required velocity, which results in the ability to create traditional control systems by shifting the centre relative to the wheels. Different combinations of modules and centre points can create Ackerman steering, differential drive and nonholonomic omni directional movement. Such centralised control can be performed by the controllers 854, for example nominating one controller as a master and others as slaves, having a centralised in built controller optionally integrated into one of the modular wheels, and/or could be performed by the processing systems 410.

[0205] Example configurations are shown in Figures 9A to 9D. Figure 9A shows a three-wheel configuration, with an instantaneous centre of rotation (ICR) placed centrally between all attached wheels producing nonholonomic omnidirectional configuration. Figure 9B shows a four-wheel configuration with an ICR placed inline with the drive axis of the rear two wheels to provide Ackerman control. Figure 9C shows a four-wheel configuration with an ICR placed inline between both sets of wheels produces differential drive or skid steer, whilst Figure 9D shows a three-wheel configuration, with an ICR inline with a drive axis of to provide tricycle control.

[0206] It will be appreciated that other drive configurations can also be employed and these are for the purpose of illustration only.

- 42 -

[0207] Examples of the processes for control movement of objects and optionally for simultaneously performing situational awareness monitoring will now be described in further detail.

[0208] For the purpose of these examples it is assumed that one or more processing systems 410 act to monitor image streams from the image devices 420, analyse the image streams to identify object locations and generate control signals that are transferred to the controllers 454, allowing the modular wheels to be controlled, to thereby move the object.

[0209] User interaction can be performed based on user inputs provided via the client devices 430, with resulting notification of model visualisations being displayed by the client devices 430. In one example, to provide this in a platform agnostic manner, allowing this to be easily accessed using client devices 430 using different operating systems, and having different processing capabilities, input data and commands are received from the client devices 430 via a webpage, with resulting visualisations being rendered locally by a browser application, or other similar application executed by the client device 430.

[0210] The processing system 410 is therefore typically a server (and will hereinafter be referred to as a server) which communicates with the client devices 430, controllers 454 and imaging devices 420, via a communications network 440, or the like, depending on the particular network infrastructure available.

[0211] To achieve this the server 410 typically executes applications software for analysing images, as well as performing other required tasks including storing and processing of data, generating control instructions, or the like, with actions performed by the server 410 being performed by the processor 511 in accordance with instructions stored as applications software in the memory 512 and/or input commands received from a user via the I/O device 513, or commands received from the client device 430.

[0212] It will also be assumed that the user interacts with the server 410 via a GUI (Graphical User Interface), or the like presented on the server 410 directly or on the client device 430, and in one particular example via a browser application that displays webpages hosted by the server 410, or an App that displays data supplied by the server 410. Actions

- 43 -

performed by the client device 430 are performed by the processor 631 in accordance with instructions stored as applications software in the memory 632 and/or input commands received from a user via the I/O device 633.

[0213] However, it will be appreciated that the above described configuration assumed for the purpose of the following examples is not essential, and numerous other configurations may be used. It will also be appreciated that the partitioning of functionality between the client devices 430, and the server 410 may vary, depending on the particular implementation.

[0214] An example of a process for controlling movement of an object having modular wheels will now be described with reference to Figure 10.

[0215] In this example, at step 1000, the server 410 receives image streams from each of the imaging devices 420, and analyses the image streams to identify synchronous overlapping images at step 1010. Specifically, this will involve analysing images from imaging devices having overlapping fields of view and taking into account timing information, such as a timestamp associated with the images and/or a time of receipt of the images, to thereby identify the synchronous images. At step 1020 the server 410 determines one or more object locations within the environment using a visual hull technique, performed using synchronous overlapping images of objects.

[0216] At step 1030, the server 410 identifies objects and/or wheels, for example using object recognition and/or detecting coded data, such as April Tags, QR codes, or similar. Having identified the object and/or modular wheels associated with the object, the server 410 can retrieve routing data associated with the object. The routing data could be a predefined route through the environment, or could include a target destination, with the server 410 then operating to calculate a route.

[0217] Following this the server 410 can generate control instructions in accordance with the route, with the control instructions can be transferred to the controllers 454, allowing the object to be moved in accordance with the routing data and thereby follow the route.

[0218] It will be appreciated that this process could be repeated periodically, such as every few seconds, allowing the server 410 to substantially continuously monitor movement of the

object, to ensure the route is followed, and to take interventions if needed, for example to account for situational awareness events, or to correct any deviation from an intended travel path. This also reduces the complexity of the control instructions needed to be generated on each loop of the control process, allowing complex movements to be implemented in as a series of simple control instructions.

[0219] A further example of a process for controlling an object and monitoring situational awareness will now be described with reference to Figures 11A and 11B.

[0220] In this example, at step 1100, the server 410 acquires multiple image streams from the imaging devices 420. At step 1105 the server 410 operates to identify objects within the image streams, typically by analysing each image stream to identify movements within the image stream. Having identified objects, at step 1110, the server 410 operates to identify synchronous overlapping images, using information regarding the relative position of the different imaging devices. At step 1115 the server 410 employs a visual hull analysis to locate objects in the environment.

[0221] Having identified locations of objects within the environment, this information is used to update an environment model at step 1120. In this regard, the environment model is a model of the environment including information regarding the current and optionally historical location of objects within the environment, which can then be used to track object movements and/or locations. An example of this will be described in more detail below.

[0222] At step 1125 the server 410 identifies movable objects and/or wheels, for example using coded data, a network address, or the like. Once the object and/or wheels have been identified, an object / wheel configuration can be determined.

[0223] In this regard, the object configuration is typically indicative of an extent of the object, but may also be indicative of additional information, such as limitations on the speed at which an object should be moved, proximity restrictions associated with the object or similar. The wheel configuration is typically indicative of a position of each wheel on the object, and is used to control the relative velocity and/or orientation of each wheel in order to generate a particular object movement.

[0224] The object and/or wheel configurations could be predefined, for example during a set-up process when the wheels are attached to the object, or could be detected for example using the visual hull analysis to work out the extent of the object, and using coded identifiers, such as April Tags on the wheels to thereby identify and locate the wheels.

[0225] At step 1135 the server 410 determines routing data defining a route and/or destination for the object. Again, this is typically predefined, for example during a set-up process, and retrieved as required, or could be manually input by an operator or other individual, for example using a client device 430.

[0226] It will be appreciated that the object and/or wheel configurations and routing data may only need to be determined a single time, such as the first time an object is detected within the environment, with relevant information being associated with the object in the environment model to allow control instructions to be generated in subsequent control loops.

[0227] It will also be appreciated that in the event that no object or wheel configurations have been determined, these could be detected. For example, a basic object and/or wheel configuration could be generated through image analysis, allowing a basic shape of the object and/or wheel position to be derived using a visual hull analysis. Wheel positions relative to each other could also be determined using coded data with fiducial markings, such as April Tags, or similar. Additionally, wheel configuration data could be derived by selectively controlling one or more of the modular wheels and monitoring resulting movement of the object and/or wheel, using this feedback to derive relative poses of the wheels, and hence generate a wheel configuration.

[0228] Assuming situational awareness monitoring is implemented, at step 1140 movement of other objects is determined and tracked. Once the object movements and/or location of static objects are known, the movements or locations are compared to situational awareness rules at step 1145, allowing situational awareness events, such as breaches of the rules, to be identified. This information is used in order to perform any required actions, such as generation of alerts or notifications, controlling of AGVs, or the like, at step 1150.

[0229] Following this, or otherwise, at step 1155 the server 410 calculates a route for the movable object. In this regard, the route will typically be generated in accordance with the routing data, but can optionally take into account a location and/or movement of other objects, as well as the extent of the object and any other relevant parameters in the object configuration, allowing the object to be moved so as to avoid the other objects.

[0230] Finally, at step 1160, the server 410 calculates control instructions to be sent to each of the modular wheels associated with the object. This is performed based on the wheel configuration, so that the server 410 can calculate necessary wheel rotations and/or orientations for each modular wheel in order to implement the respective route. The control instructions are then provided to each wheel controller 454, allowing the wheels to be moved accordingly so that the object follows the calculated route.

[0231] As mentioned above, the above described approach typically relies on a calibration process, which involves calibrating the imaging devices 420 both intrinsically, in order to determine imaging device properties, and extrinsically, in order to take into account positions of the imaging devices within the environment. An example of such a calibration process will now be described with reference to Figure 12.

[0232] In this example, at step 1200 the imaging devices are used to capture images of patterns. The patterns are typically of a predetermined known form and could include patterns of dots, machine readable coded data, such as April Tags, or the like. The images of the patterns are typically captured from a range of different angles.

[0233] At step 1210 the images are analysed by comparing the captured images to reference images representing the intended appearance of the known patterns, allowing results of the comparison to be used to determine any distortions or other visual effects arising from the characteristics of the particular imaging device. This is used to derive intrinsic calibration data for the imaging device at step 1220, which is then stored as part of calibration data, allowing this to be used to correct images captured by the respective imaging device, so as to generate a corrected image.

[0234] It will be appreciated that steps 1200 to 1220 are repeated for each individual imaging device to be used, and can be performed in situ, or prior to placement of the imaging devices.

[0235] At step 1230, assuming this has not already been performed, the cameras and one or more targets are positioned in the environment. The targets can be of any appropriate form and could include dots, fiducial markings such as April Tags, or the like.

[0236] At step 1240 images of the targets are captured by the imaging devices 420, with the images being provided to the server 410 for analysis at step 1250. The analysis is performed in order to identify targets that have been captured by the different imaging devices 420 from different angles, thereby allowing the relative position of the imaging devices 420 to be determined. This process can be performed manually, for example by having the user highlight common targets in different images, allowing triangulation to be used to calculate the location of the imaging devices that captured the images. Alternatively, this can be performed at least in part using image processing techniques, such as by recognising different targets positioned throughout the environment, and then again using triangulation to derive the camera positions. This process can also be assisted by having the user identify an approximate location of the cameras, for example by designating these within an environment model.

[0237] At step 1260 extrinsic calibration data indicative of the relative positioning of the imaging devices is stored as part of the calibration data.

[0238] An example of a process for performing situational monitoring will now be described in more detail with reference to Figures 13A to 13C.

[0239] In this example, at step 1300 image streams are captured by the imaging devices 410 with these being uploaded to the server 410 at step 1302. Steps 1300 and 1302 are repeated substantially continuously so that image streams are presented to the server 410 substantially in real time.

[0240] At step 1304 image streams are received by the server 410, with the server operating to identify an image capture time at step 1306 for each image in the image stream, typically based on a time stamp associated with each image and provided by the respective imaging

device 420. This time is then optionally validated at step 1308, for example by having the server 410 compare the time stamped capture time to a time of receipt of the images by the server 410, taking into account an expected transmission delay, to ensure that the times are within a predetermined error margin. In the event that the time is not validated, an error can be generated allowing the issue to be investigated.

[0241] Otherwise, at step 1310 the server 410 analyses successive images and operates to subtract static regions from the images at step 1312. This is used to identify moving components within each image, which are deemed to correspond to objects moving within the environment.

[0242] At step 1314, synchronous overlapping images are identified by identifying images from different image streams that were captured substantially simultaneously, and which include objects captured from different viewpoints. Identification of overlapping images can be performed using the extrinsic calibration data, allowing cameras with overlapping field of view to be identified, and can also involve analysis of images including objects to identify the same object in the different images. This can examine the presence of machine readable coded data, such as April Tags within the image, or can use recognition techniques to identify characteristics of the objects, such as object colours, size, shape, or the like.

[0243] At step 1316, the images are analysed to identify object locations at step 1318. This can be performed using coded fiducial markings, or by performing a visual hull analysis, in the event that such markings are not available. It will be appreciated that in order to perform the analysis this must take into account the extrinsic and intrinsic calibration data to correct the images for properties of the imaging device, such as any image distortion, then further utilising knowledge of the relative position of the respective imaging devices in order to interpret the object location and/or a rough shape in the case of performing a visual hull analysis.

[0244] Having determined an object location, at step 1320 the server 410 operates to determine an object identity. In this regard the manner in which an object is identified will vary depending on the nature of the object, and any identifying data, and an example identification process will now be described in more detail with reference to Figure 14.

[0245] In this example, at step 1400, the server 410 analyses one or more images of the object, using image processing techniques, and ascertains whether the image includes visual coded data, such as an April Tag at step 1405. If the server identifies an April Tag or other visual coded data, this is analysed to determine an identifier associated with the object at step 1410. An association between the identifier and the object is typically stored as object data in a database when the coded data is initially allocated to the object, for example during a set-up process when an April tag is attached to the object. Accordingly, decoding the identifier from the machine readable coded data allows the identity of the object to be retrieved from the stored object data, thereby allowing the object to be identified at step 1415.

[0246] In the event that visual coded data is not present, the server 410 determines if the object is coincident with a reader, such as a Bluetooth or RFID tag reader at step 1420. If so, the tag reader is queried at step 1425 to ascertain whether tagged data has been detected from a tag associated with the object. If so, the tag data can be analysed at step 1410 to determine an identifier, with this being used to identify the object at step 1415, using stored object data in a manner similar to that described above.

[0247] It will be appreciated that the above described techniques can also be used for identifying individual wheels in the event that the wheels are associated with respective identifiers and/or tags.

[0248] In the event that tag data is not detected, or the object is not coincident with the reader, a visual analysis can be performed using image recognition techniques at step 1435 in order to attempt to identify the object at step 1440. It will be appreciated that this may only be sufficient to identify a type of object, such as a person, and might not allow discrimination between objects of the same type. Additionally, in some situations, this might not allow for identification of objects, in which case the objects can be assigned an unknown identity.

[0249] At step 1322 the server 410 accesses an existing environment model and assesses whether a detected object is a new object at step 1324. For example, if a new identifier is detected this will be indicative of a new object. Alternatively, for objects with no identifier, the server 410 can assess if the object is proximate to an existing object within the model, meaning the object is an existing object that has moved. It will be noted in this regard, that

as objects are identified based on movement between images, an object that remains static for a prolonged period of time may not be detected within the images, depending on how the detection is performed as previously described. However, a static object will remain present in the environment model based on its last known location, so that when the object recommences movement, and is located within the images, this can be matched to the static object within the environment model, based on the coincident location of the objects, although it will be appreciated that this may not be required depending on how objects are detected.

[0250] If it is determined that the object is a new object, the object is added to the environment model at step 1326. In the event that the object is not a new object, for example if it represents a moved existing object, the object location and/or movement can be updated at step 1328.

[0251] In either case, at step 1330, any object movement can be extrapolated in order to predict future object movement. Thus, this will examine a trend in historical movement patterns and use this to predict a likely future movement over a short period of time, allowing the server 410 to predict where objects will be a short time in advance. As previously described, this can be performed using machine learning techniques or similar, taking into account previous movements for the object, or objects of a similar type, as well as other information, such as defined intended travel paths.

[0252] At step 1332, the server 410 retrieves rules associated with each object in the environment model using the respective object identity. The rules will typically specify a variety of situational awareness events that might arise for the respective object, and can include details of permitted and/or denied movements. These could be absolute, for example, comparing an AGV's movement to a pre-programmed travel path, to ascertain if the object is moving too far from the travel path, or comparing an individual's movement to confirm they are within permitted zones, or outside denied zones. The situational awareness rules could also define relative criteria, such as whether two objects are within less than a certain distance of each other, or are on intersecting predicted travel paths.

[0253] As previously described, the rules are typically defined based on an understanding of requirements of the particular environment and the objects within the environment. The rules are typically defined for specific objects, object types or for multiple objects, so that different rules can be used to assess situational awareness for different objects. The situational awareness rules are typically stored together with associated object identities, either in the form of specific object identifiers, or specified object types, as rule data, allowing the respective rules to be retrieved for each detected object within the environment.

[0254] At step 1334 the rules are applied to the object location, movement or predicted movement associated with each object, to determine if the rules have been breached at step 1336, and hence that an event is occurring.

[0255] If it is assessed that the rules are breached at step 1336, the server 410 determines any action required at step 1338, by retrieving the action from the rules data, allowing the action to be initiated at step 1340. In this regard, the relevant action will typically be specified as part of the rules, allowing different actions to be defined associated with different objects and different situational awareness events. This allows a variety of actions to be defined, as appropriate to the particular event, and this could include, but is not limited to, recording an event, generating alerts or notifications, or controlling autonomous or semi-autonomous vehicles.

[0256] In one example, client devices 430, such as mobile phones, can be used for displaying alerts, so that alerts could be broadcast to relevant individuals in the environment. This could include broadcast notifications pushed to any available client device 430, or could include directing notifications to specific client devices 430 associated with particular users. For example, if an event occurs involving a vehicle, a notification could be provided to the vehicle operator and/or a supervisor. In a further example, client devices 430 can include displays or other output devices, such as beacons configured to generate audible and/or visual alerts, which can be provided at specific defined locations in the environment, or associated with objects, such as AGVs. In this instance, if a collision with an AGV or other object is imminent, a beacon on the object can be activated alerting individuals to the potential collision, and thereby allowing the collision to be avoided.

[0257] In a further example, the client device 430 can form part of, or be coupled to, a control system of an object such as an autonomous or semi-autonomous vehicle, allowing the server 410 to instruct the client device 430 to control the object, for example causing movement of the object to cease until the situational awareness event is mitigated.

[0258] It will be appreciated from the above that client devices 430 can be associated with respective objects, or could be positioned within the environment, and that this will be defined as part of a set-up process. For example, client devices 430 associated with objects could be identified in the object data, so that when an action is to be performed associated with a respective object, the object data can be used to retrieve details of the associated client device and thereby push notifications to the respective client device. Similarly, details of static client devices could be stored as part of the environment model, with details being retrieved in a similar manner as needed.

[0259] In any event, once the action has been initiated or otherwise, the process can return to step 1304 to allow monitoring to continue.

[0260] Whilst the above described example has focused on situational awareness monitoring, it will be appreciated that this can be performed in conjunction with controlling a movable object, specifically, in this example routing data and information regarding object movement and locations can be used to control the modular wheels and hence control movement of the object.

[0261] In addition to performing situational awareness monitoring, and performing actions as described above, the server 410 also maintains the environment model and allows this to be viewed by way of a graphical representation. This can be achieved via a client device 430, for example, allowing a supervisor or other individual to maintain an overview of activities within the environment, and also potentially view situational awareness events as they arise. An example of a graphical representation of an environment model will now be described in more detail with reference to Figure 15.

[0262] In this example the graphical representation 1500 includes an environment *E*, such as an internal plan of a building or similar. It will be appreciated that the graphical

representation of the building can be derived based on building plans and/or by scanning or imaging the environment. The model includes icons 1520 representing the location of imaging devices 420, icons 1501, 1502, 1503 and 1504 representing the locations of objects, and icons 1530 representing the location of client devices 430, such as beacons used in generating audible and/or visual alerts.

[0263] The positioning of imaging devices 420 can be performed as part of the calibration process described above, and may involve having a user manually position icons at approximate locations, with the positioning being refined as calibration is performed. Similarly, positioning of the client devices could also be performed manually, in the case of static client devices 420, and/or by associating a client device 420 with an object, so that the client device location is added to the model when the object is detected within the environment.

[0264] In this example, the representation also displays additional details associated with objects. In this case, the object 1501 is rectangular in shape, which could be used to denote an object type, such as an AGV, with the icon having a size similar to the footprint of the actual physical AGV. The object 1501 has an associated identifier ID1501 shown, which corresponds to the detected identity of the object. In this instance, the object has an associated pre-programmed travel path 1501.1, which is the path the AGV is expected to follow, whilst a client device icon 1530.1 is shown associated with the object 1501, indicating that a client device is provided on the AGV.

[0265] In this example, the object 1502 is a person, and hence denoted with a different shape to the object 1501, such as a circle, again having a footprint similar to that of the person. The object has an associated object identifier IDPer2, indicating this is a person, and using a number to distinguish between different people. The object 1502 has a travel path 1502.2, representing historical movement of the object 1502 within the environment.

[0266] Object 1503 is again a person, and includes an object identifier IDPer3, to distinguish from the object 1502. In this instance the object is static and is as a result shown in dotted lines.

[0267] Object 1504 is a second AGV, in this instance having an unknown identifier represented by ID???. The AGV 1504 has a historical travel path 1504.2 and a predicted travel path 1004.3. In this instance it is noted that the predicted travel path intersects with the predetermined path 1501.1 of the AGV 1501 and it is anticipated that an intersection may occur in the region 1504.4, which is highlighted as being a potential issue.

[0268] Finally a denied region 1505 is defined, into which no objects are permitted to enter, with an associated client device 430 being provided as denoted by the icon 1530.5, allowing alerts to be generated if objects approach the denied region.

[0269] As described above, image regions are classified in order to assess whether the image region is of a static part of the environment, or part of the environment including movement, which can in turn be used to identify objects. An example of a process for classifying image regions will now be described in more detail with reference to Figure 16.

[0270] At step 1600, an image region is identified. The image region can be arbitrarily defined, for example by segmenting each image based on a segmentation grid, or similar, or may be based on the detection of movement in the previous images. Once an image region is identified, the server optionally assesses a history of the image region at step 1610, which can be performed in order to identify if the image region has recently been assessed as non-static, which is in turn useful in identifying objects that have recently stopped moving.

[0271] At step 1620, visual effects can be identified. Visual effects can be identified in any appropriate manner depending on the nature of the visual effect and the preferred implementation. For example, this may involve analysing signals from illumination sensors to identify changes in background or ambient illumination. Alternatively this could involve retrieving information regarding visual effect locations within the environment, which could be defined during a calibration process, for example by specifying the location of screens or displays in the environment. This can also involve analysing the images in order to identify visual effects, for example to identify parts of the image including a spectral response known to correspond to a visual effect, such as a particular illumination source. This process is performed to account for visual effects and ensure these are not incorrectly identified as moving objects.

[0272] At step 1630, a classification threshold is set, which is used to assess whether an image region is static or non-static. The classification threshold can be a default value, which is then modified as required based on the image region history and/or identified visual effects. For example, if the individual region history indicates that the image region was previously or recently classified as non-static, the classification threshold can be raised from a default level to reduce the likelihood of the image region are being classified as static in the event that an object has recently stopped moving. This in effect increases the learning duration for assessing changes in the respective image region, which is useful in tracking temporarily stationary objects. Similarly, if visual effects are present within the region, the threat classification threshold can be modified to reduce the likelihood of the image region are being misclassified.

[0273] Once the classification threshold has been determined, changes in that the image region are analysed at step 1640. This is typically performed by comparing the same image region across multiple images of the image stream. The multiple images can be successive images but this is not essential and any images which are temporally spaced can be assessed. Following this, at step 1650 the changes in the image region are compared to a threshold, with results of the comparison being used to classify the image region at step 1660, for example a defining the region to be a static if the degree of movement falls below the classification threshold.

[0274] As previously described, occlusions may arise in which objects are at least partially obstructed from an imaging device, and an example of a process for occlusion detection and mitigation will now be described with reference to Figure 17.

[0275] In this example, corresponding image regions are identified in multiple overlapping images at step 1700. In this regard, the corresponding image regions are image regions from multiple overlapping images that are a view of a common volume in the environment, such as one or more voxels. At step 1710 one or more candidate objects are identified, for example using a visual hull technique.

[0276] At step 1720, any candidate objects are added to a three-dimensional model, such as a model similar to that described above with respect to Figure 10. At step 1730, the candidate

objects are back projected onto the imaging plane of an imaging device that captured an image of the candidate objects. This is performed in order to ascertain whether the candidate objects are overlapping and hence an occlusion may have occurred.

[0277] Once potential occlusions have been identified, at step 1740 the server can use this information to validate candidate objects. For example, for candidate objects not subject to occlusion, these can be accepted to detected objects. Conversely, where occlusions are detected, the visual hull process can be repeated taking the occlusion into account. This could be achieved by removing the image region containing the occlusion from the visual hull process, or more typically taking the presence of the occlusion into account, for example using a weighting process or similar as will be described in a more detail below.

[0278] An example of weighting process for object identification will now be described with reference to Figure 18.

[0279] In this example, at step 1800 corresponding image regions are identified in a manner similar to that described with respect to step 1200.

[0280] At step 1810, each image region is assessed, with the assessment being used to ascertain the likelihood that the detection of an object is accurate. In this regard, it will be appreciated that successful detection of an object will be influenced by a range of factors, including image quality such as image resolution or distortion, camera geometry such as a camera distance and angle, image region history such as whether the image region was previous static or non-static, the presence or absence of occlusions or visual effects, a degree of asynchronicity between collected data, such as difference in capture time of the overlapping images, or the like.

[0281] Accordingly, this process attempts to take these factors into account by assigning a value based on each factor, and using this to determine an image region score at step 1820. For example, the value for each factor will typically represent whether the factor will positively or negatively influence successful detection of an object so if an occlusion is at present a value of -1 could be used, whereas if an occlusion is not present a value of +1 could

be used, indicating that it is more likely an object detection would be correct than if an occlusion is present.

[0282] Once calculated, the image region scores can then be used in the identification of objects.

[0283] In one example, at step 1830 the visual hull process is performed, using the image region score as a weighting. Thus, in this instance, an image region with a low image region score, which is less likely to have accurately imaged the object, will be given a low weighting in the visual hull process. Consequently, this will have less influence on the object detection process, so the process is more heavily biased to image regions having a higher image region source. Additionally and/or alternatively, a composite object score can be calculated by combining the image region score of each of the corresponding image regions at step 1840, with the resulting value being compared to a threshold at step 1850, with this being used to assess whether an object has been successfully identified at step 1860.

[0284] Accordingly, it will be appreciated that the above described system operates to track movement of objects within the environment which can be achieved using low cost sensors. Movement and/or locations of the objects can be compared to defined situational awareness rules to identify rule breaches which in turn can allow actions to be taken such as notifying of the breach and/or controlling AGVs in order to prevent accidents or other compliance event occurring.

[0285] Throughout this specification and claims which follow, unless the context requires otherwise, the word “comprise”, and variations such as “comprises” or “comprising”, will be understood to imply the inclusion of a stated integer or group of integers or steps but not the exclusion of any other integer or group of integers. As used herein and unless otherwise stated, the term "approximately" means $\pm 20\%$.

[0286] Persons skilled in the art will appreciate that numerous variations and modifications will become apparent. All such variations and modifications which become apparent to persons skilled in the art, should be considered to fall within the spirit and scope that the invention broadly appearing before described.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

- 1) A system for moving an object within an environment, wherein the system includes:
 - a) at least one modular wheel configured to move the object, wherein the at least one modular wheel includes:
 - i) a body configured to be attached to the object;
 - ii) a wheel;
 - iii) a drive configured to rotate the wheel; and,
 - iv) a controller configured to be attached to the body and to control the drive; and,
 - b) one or more processing devices configured to:
 - i) receive an image stream including a plurality of captured images from each of a plurality of imaging devices positioned in the environment, the plurality of imaging devices being configured to capture images of the object within the environment;
 - ii) analyse the images to determine an object location within the environment;
 - iii) generate control instructions at least in part using the determined object location; and,
 - iv) provide the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.
- 2) A system according to claim 1, wherein the system includes one or more passive wheels mounted to the object.
- 3) A system according to claim 1 or claim 2, wherein the at least one modular wheel includes a steering drive configured to adjust an orientation of the wheel, and wherein the controller is configured to control the steering drive to thereby change an orientation of the wheel.
- 4) A system according to any one of the claims 1 to 3, wherein the at least one modular wheel includes a transceiver configured to communicate wirelessly with the one or more processing devices.
- 5) A system according to any one of the claims 1 to 4, wherein the at least one modular wheel includes a power supply configured to power at least one of:
 - a) the drive;
 - b) the controller;
 - c) a transceiver; and,
 - d) a steering drive.

- 6) A system according to any one of the claims 1 to 5, wherein the control instructions include at least one of:
 - a) a wheel orientation for each wheel; and,
 - b) a rate of rotation for each wheel.
- 7) A system according to any one of the claims 1 to 6, wherein the system includes a plurality of modular wheels.
- 8) A system according to claim 7, wherein the one or more processing devices are configured to provide respective control instructions to each controller to thereby independently control each modular wheel.
- 9) A system according to claim 7, wherein the one or more processing devices are configured to provide control instructions to the controllers and wherein the controllers communicate to independently control each modular wheel.
- 10) A system according to claim 9, wherein the control instructions include a direction and rate of travel for the object, and wherein the controllers use the control instructions to determine at least one of:
 - a) a wheel orientation for each wheel; and,
 - b) a rate of rotation for each wheel.
- 11) A system according to any one of the claims 1 to 10, wherein the system is configured to steer the vehicle by at least one of:
 - a) differentially rotating multiple modular wheels; and,
 - b) changing an orientation of one or more modular wheels.
- 12) A system according to any one of the claims 1 to 11, wherein the one or more processing devices are configured to:
 - a) determine an object configuration; and,
 - b) generate the control instructions at least partially in accordance with the object extent.
- 13) A system according to claim 12, wherein the object configuration is indicative of at least one of:
 - a) a physical extent of the object; and,
 - b) movement parameters associated with the object.
- 14) A system according to any one of the claims 1 to 13, wherein the one or more processing devices are configured to:

- a) determine a wheel configuration indicative of a position of each wheel relative to the object; and,
 - b) generate the control instructions at least partially in accordance with the wheel configuration.
- 15) A system according to any one of the claims 1 to 14, wherein the at least one modular wheel are each attached to the object at known locations.
- 16) A system according to any one of the claims 1 to 15, wherein the object includes a platform and wherein the at least one modular wheel is attached to the platform.
- 17) A system according to any one of the claims 1 to 16, wherein the object includes an item supported by the platform.
- 18) A system according to any one of the claims 1 to 17, wherein the one or more processing devices are configured to:
- a) determine an identity of at least one of:
 - i) each modular wheel; and,
 - ii) the object; and,
 - b) generate control instructions in accordance with the identity.
- 19) A system according to claim 18, wherein the one or more processing devices are configured to determine the identity at least in part using a network identifier.
- 20) A system according to claim 18 or claim 19, wherein the one or more processing devices are configured to determine the identity using machine readable coded data.
- 21) A system according to claim 20, wherein the machine readable coded data is visible data, and wherein the one or more processing devices are configured to analyse the images to detect the machine readable coded data.
- 22) A system according to claim 20 or claim 21, wherein the machine readable coded data is encoded on a tag, and wherein the one or more processing devices are configured to receive signals indicative of the machine readable coded data from a tag reader.
- 23) A system according to claim 22, wherein the tags at least one of:
- a) short range wireless communications protocol tags;
 - b) RFID tags; and,
 - c) Bluetooth tags.

- 24) A system according to any one of the claims 1 to 23, wherein the one or more processing devices are configured to:
- a) determine routing data indicative of at least one of:
 - i) a travel path; and,
 - ii) a destination;
 - b) generate control instructions in accordance with the routing data and the object location.
- 25) A system according to claim 24, wherein the routing data is indicative of at least one of:
- a) a permitted object travel path;
 - b) permitted object movements;
 - c) permitted proximity limits for different objects;
 - d) permitted zones for objects;
 - e) denied zones for objects.
- 26) A system according to claim 24 or claim 25, wherein the one or more processing devices are configured to:
- a) determine an identity for at least one of:
 - i) the object; and,
 - ii) for at least one modular wheel attached to the object;
 - b) determine the routing data at least in part using the object identity.
- 27) A system according to any one of the claims 1 to 26, wherein the one or more processing devices are configured to:
- a) generate calibration control instructions;
 - b) monitor movement of at least one of the object and the at least one modular wheel in response to the calibration control instructions; and
 - c) use results of the monitoring to generate control instructions.
- 28) A system according to any one of the claims 1 to 27, wherein the imaging devices are at least one of:
- a) positioned within the environment at fixed locations; and,
 - b) static relative to the environment.
- 29) A system according to any one of the claims 1 to 28, wherein at least some of the imaging devices are positioned within the environment to have at least partially overlapping fields of view and wherein the one or more processing devices are configured to:

- a) identify overlapping images in the different image streams, the overlapping images being images captured by imaging devices having overlapping fields of view; and,
 - b) analyse the overlapping images to determine object locations within the environment.
- 30) A system according to any one of the claims 1 to 29, wherein at least some of the imaging devices are positioned within the environment to have at least partially overlapping fields of view and wherein the one or more processing devices are configured to:
- a) analyse changes in the object locations over time to determine object movements within the environment;
 - b) compare the object movements to situational awareness rules; and,
 - c) use results of the comparison to identify situational awareness events.
- 31) A system according to claim 29 or claim 30, wherein the overlapping images are synchronous overlapping images captured at approximately the same time.
- 32) A system according to claim 31, wherein the one or more processing devices are configured to:
- a) determine a capture time of each captured image; and,
 - b) identify synchronous images using the captured time.
- 33) A system according to claim 32, wherein the one or more processing devices are configured to determine a capture time using at least one:
- a) a capture time generated by the imaging device;
 - b) a receipt time associated with each image, the receipt time being indicative of a time of receipt by the one or more processing devices; and,
 - c) a comparison of image content in the images.
- 34) A system according to any one of the claims 1 to 33, wherein the one or more processing devices are configured to:
- a) analyse images from each image stream to identify object images, the object images being images including objects; and,
 - b) identify overlapping images as object images that include the same object.
- 35) A system according to claim 34, wherein the one or more processing devices are configured to identify overlapping images based at least in part on a positioning of the imaging devices.
- 36) A system according to any one of the claims 1 to 35, wherein the one or more processing devices are configured to:

- a) analyse a number of images from an image stream to identify static image regions; and,
 - b) identifying object images as images including non-static image regions.
- 37) A system according to claim 36, wherein at least one of the images is a background reference image.
- 38) A system according to any one of the claims 1 to 37, wherein the one or more processing devices are configured to determine an object location using at least one of:
- a) a visual hull technique; and,
 - b) detection of fiducial markings in the images; and,
 - c) detection of fiducial markings in multiple triangulated images.
- 39) A system according to any one of the claims 1 to 38, wherein the one or more processing devices are configured to interpret the images in accordance with calibration data.
- 40) A system according to claim 39, wherein the calibration data includes at least one of:
- a) intrinsic calibration data indicative of imaging properties of each imaging device; and,
 - b) extrinsic calibration data indicative of relative positioning of the imaging devices within the environment.
- 41) A system according to claim 39 or claim 40, wherein the one or more processing devices are configured to generate calibration data during a calibration process by:
- a) receiving images of defined patterns captured from different positions using an imaging device; and,
 - b) analysing the images to generate calibration data indicative of a image capture properties of the imaging device.
- 42) A system according to any one of the claims 39 to 41, wherein the one or more processing devices are configured to generate calibration data during a calibration process by:
- a) receiving captured images of targets within the environment;
 - b) analysing the captured images to identify images captured by different imaging devices which show the same target; and,
 - c) analysing the identified images to generate calibration data indicative of a relative position and orientation of the imaging devices.
- 43) A system according to any one of the claims 1 to 42, wherein the one or more processing devices are configured to generate an environment model, the environment model being indicative of at least one of:

- a) the environment;
 - b) a location of imaging devices in the environment;
 - c) current object locations;
 - d) object movements;
 - e) predicted obstacles;
 - f) predicted object locations; and,
 - g) predicted object movements.
- 44) A system according to claim 43, wherein the one or more processing devices are configured to generate a graphical representation of the environment model.
- 45) A system according to any one of the claims 1 to 44, wherein the one or more processing devices are configured to:
- a) analyse changes in object locations over time to determine object movements within the environment;
 - b) compare the object movements to situational awareness rules; and,
 - c) use results of the comparison to identify situational awareness events.
- 46) A system according to claim 45, wherein in response to identification of a situational awareness event, the one or more processing devices are configured to perform an action including at least one of:
- a) record an indication of the situational awareness event;
 - b) generate a notification indicative of the situational awareness event;
 - c) cause an output device to generate an output indicative of the situational awareness event;
 - d) activate an alarm; and,
 - e) cause operation of an object to be controlled.
- 47) A system according to claim 45 or claim 46, wherein the one or more processing devices are configured to:
- a) identify the situational awareness event substantially in real time; and,
 - b) perform an action substantially in real time.
- 48) A system according to any one of the claims 1 to 47, wherein the imaging devices are at least one of:
- a) security imaging devices;

- b) monoscopic imaging devices;
 - c) non-computer vision based imaging devices; and,
 - d) imaging devices that do not have associated intrinsic calibration information.
- 49) A method for moving an object within an environment, the method being performed using a system including:
- a) at least one modular wheel configured to move the object, wherein the at least one modular wheel each includes:
 - i) a body configured to be attached to the object;
 - ii) a wheel;
 - iii) a drive configured to rotate the wheel; and,
 - iv) a controller configured be attached to the body and to to control the drive; and,
 - b) one or more processing devices, wherein the method includes, in the one or more processing devices:
 - i) receiving an image stream including a plurality of captured images from each of a plurality of imaging devices positioned in the environment, the plurality of imaging devices being configured to capture images of the object within the environment;
 - ii) analysing the images to determine an object location within the environment;
 - iii) generating control instructions at least in part using the determined object location; and,
 - iv) providing the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.
- 50) A computer program product for moving an object within an environment using a system including:
- a) at least one modular wheel configured to move the object, wherein the at least one modular wheel each includes:
 - i) a body configured to be attached to the object;
 - ii) a wheel;
 - iii) a drive configured to rotate the wheel; and,
 - iv) a controller configured to be attached to the body and to control the drive; and,

- b) one or more processing devices, wherein the computer program product includes computer executable code, which when executed by the one or more processing devices causes the one or more processing devices to:
 - i) receive an image stream including a plurality of captured images from each of a plurality of imaging devices positioned in the environment, the plurality of imaging devices being configured to capture images of the object within the environment;
 - ii) analyse the images to determine an object location within the environment;
 - iii) generate control instructions at least in part using the determined object location; and,
 - iv) provide the control instructions to the controller, the controller being responsive to the control instructions to control the drive and thereby move the object.

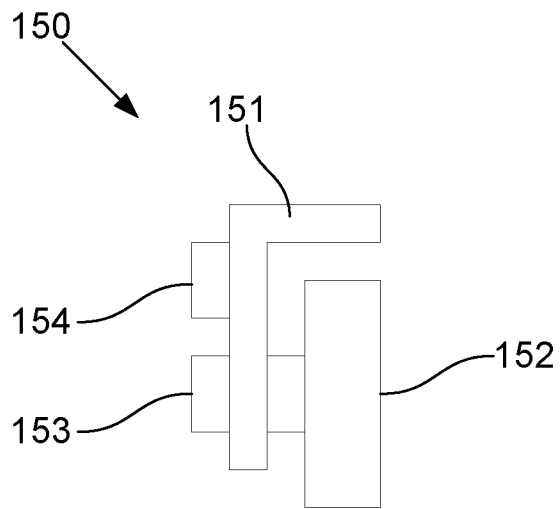


Fig. 1A

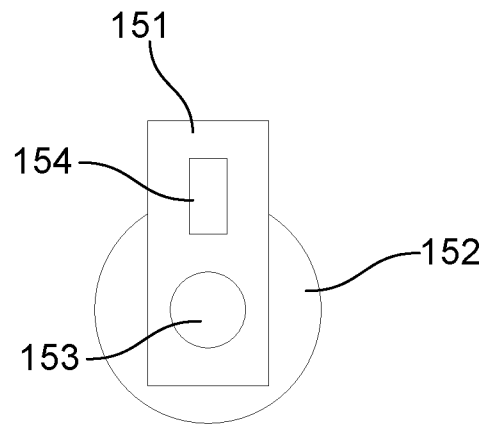


Fig. 1B

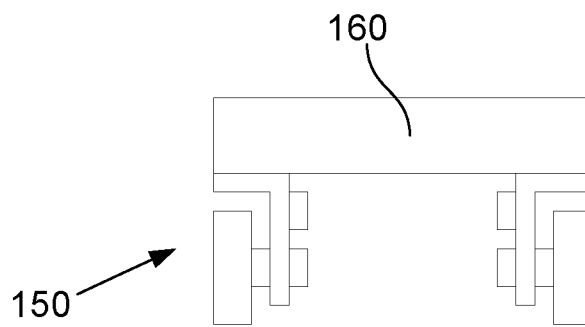


Fig. 1C

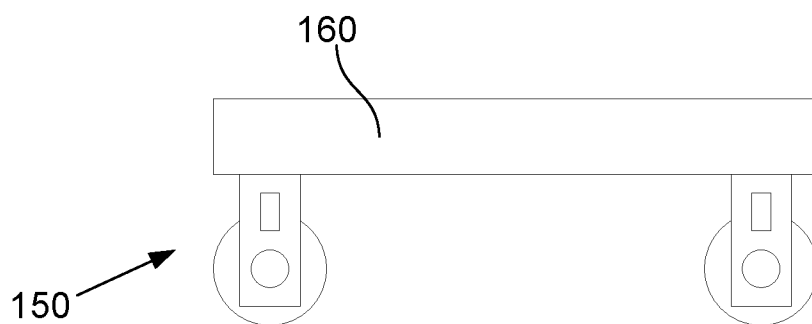
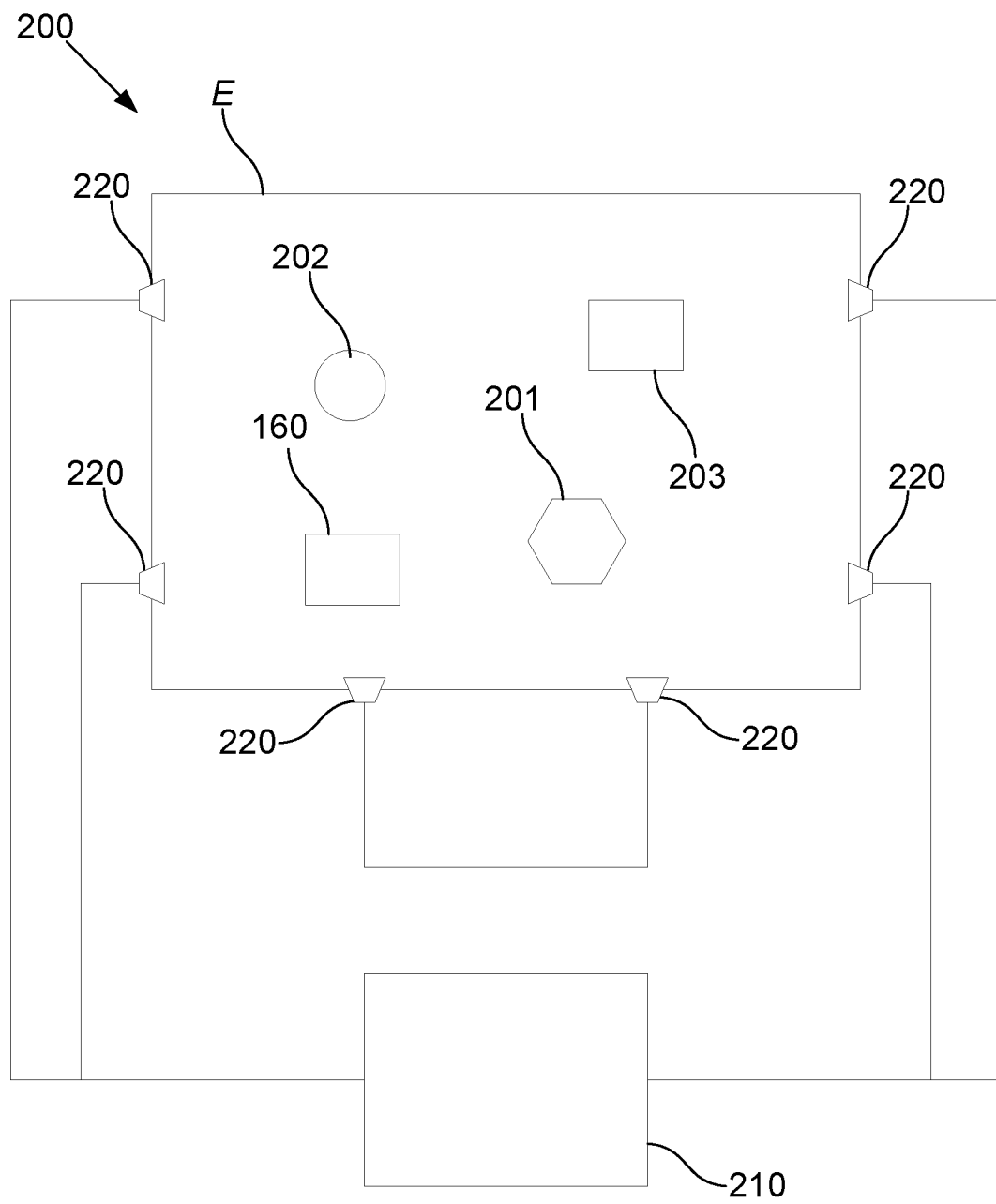


Fig. 1D

**Fig. 2**

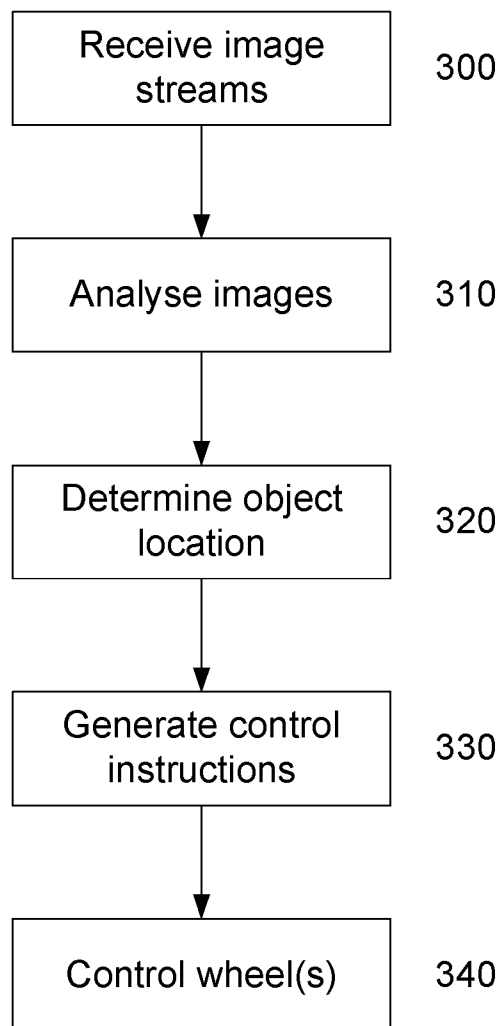


Fig. 3

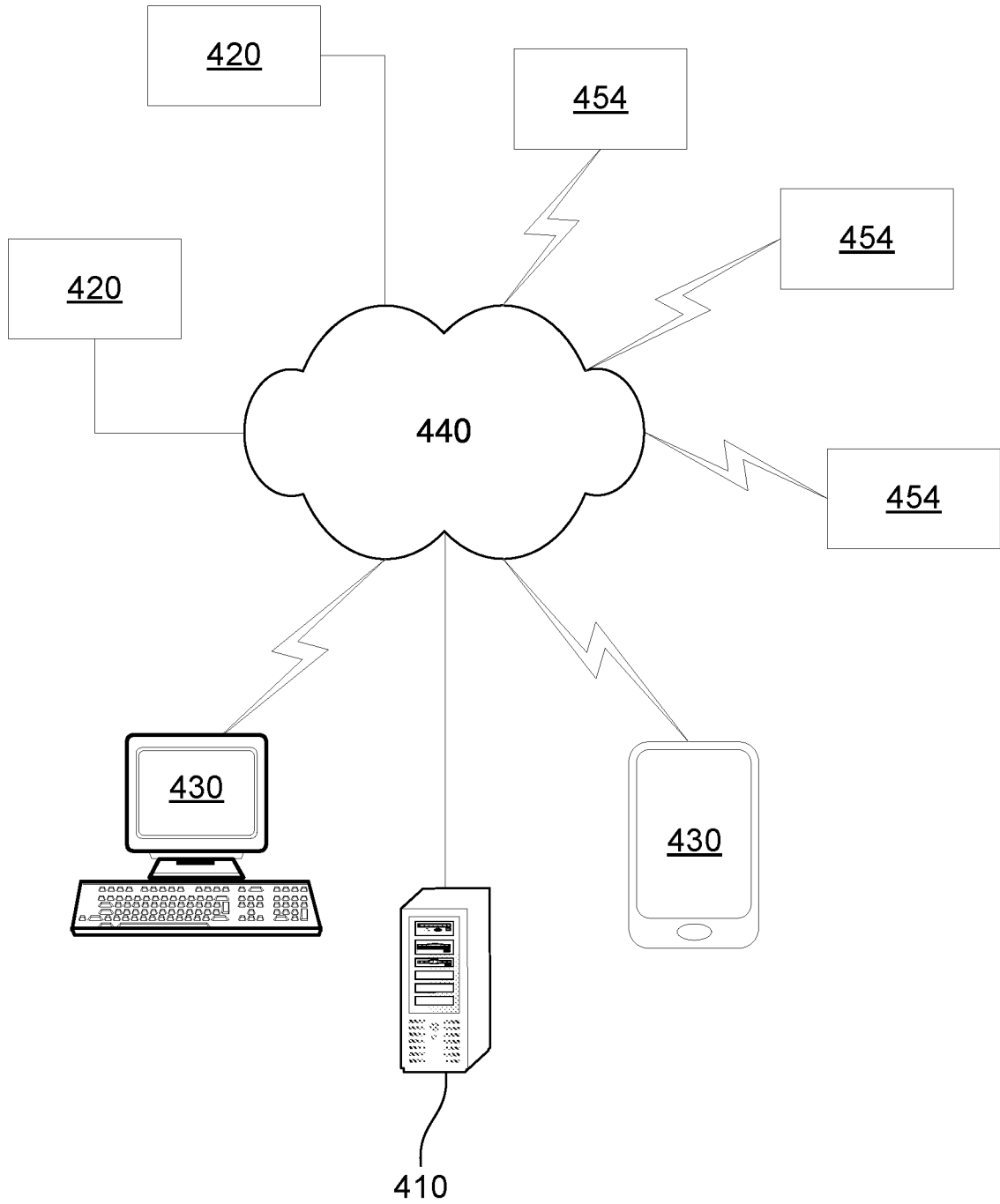


Fig. 4

410

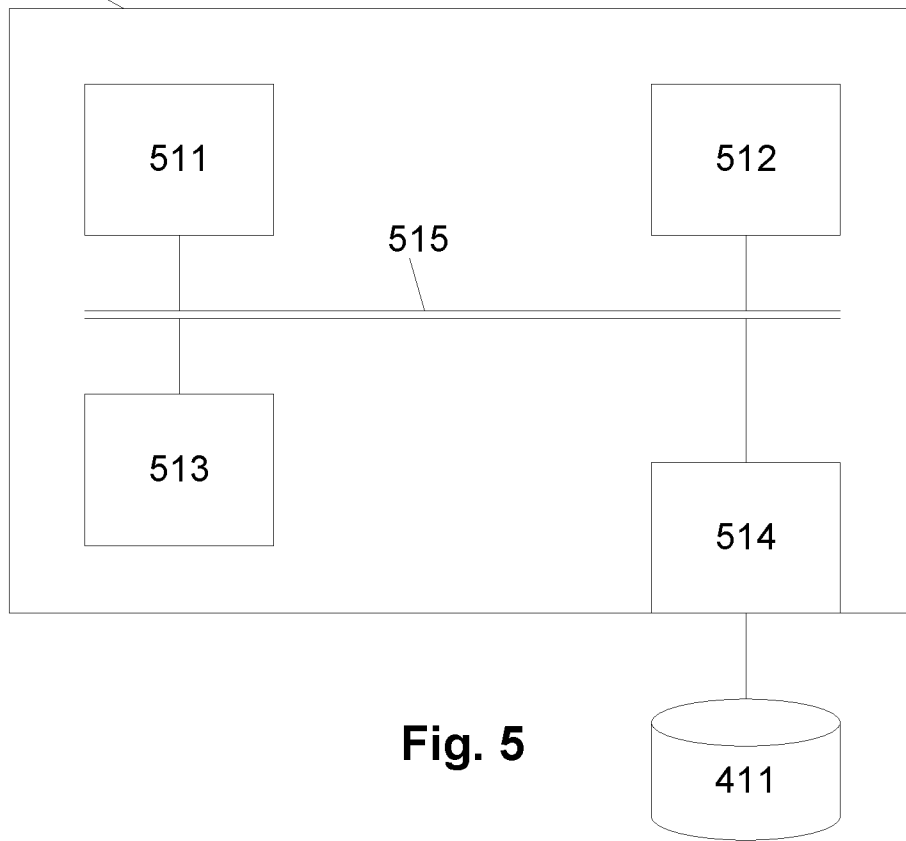


Fig. 5

430

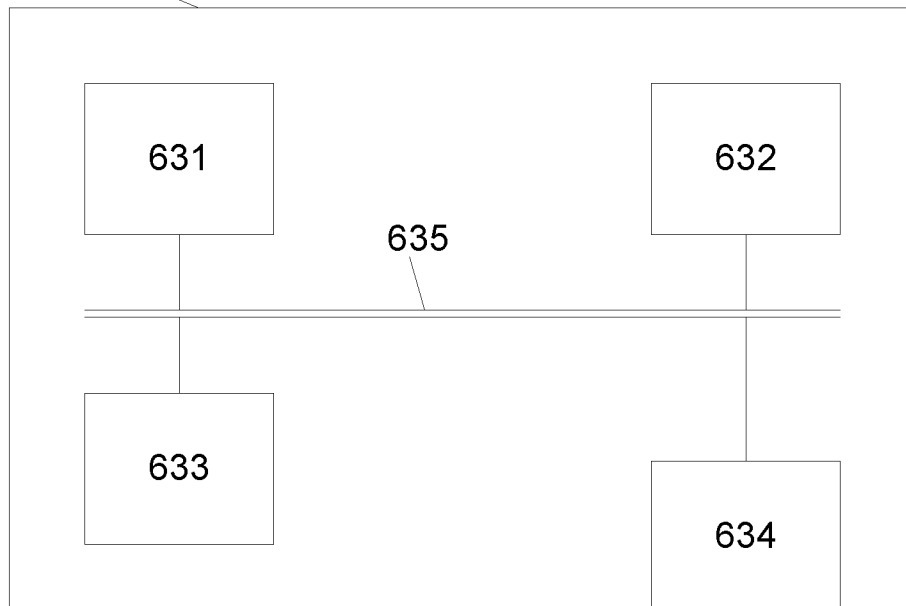


Fig. 6

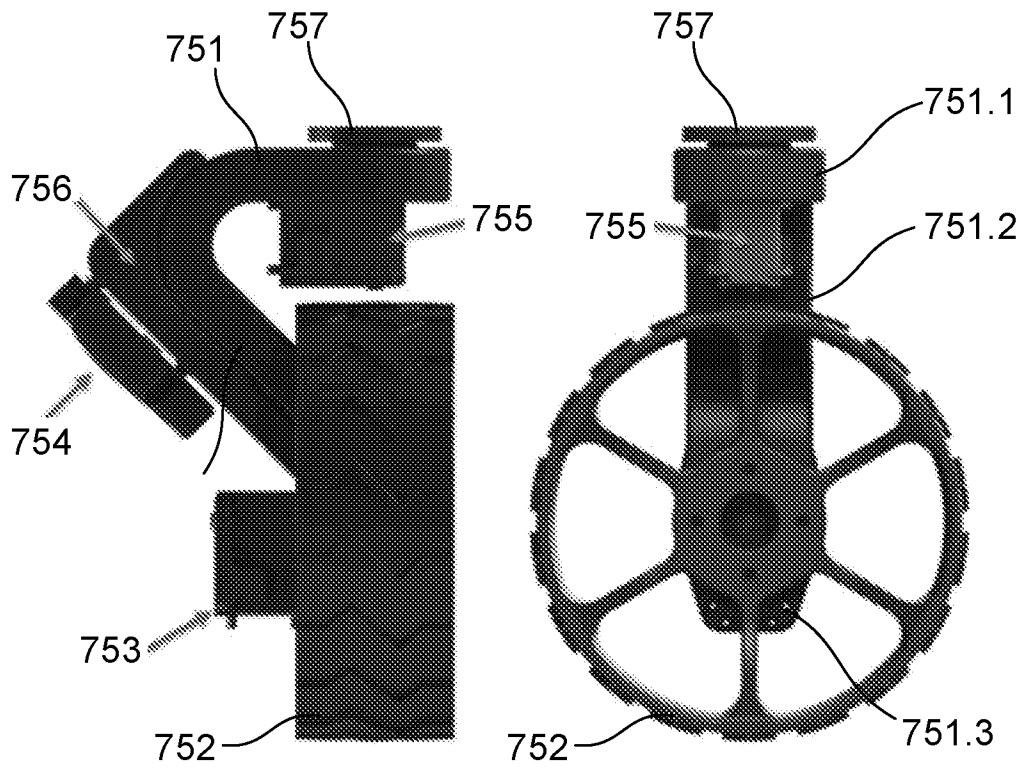


Fig. 7A

Fig. 7B

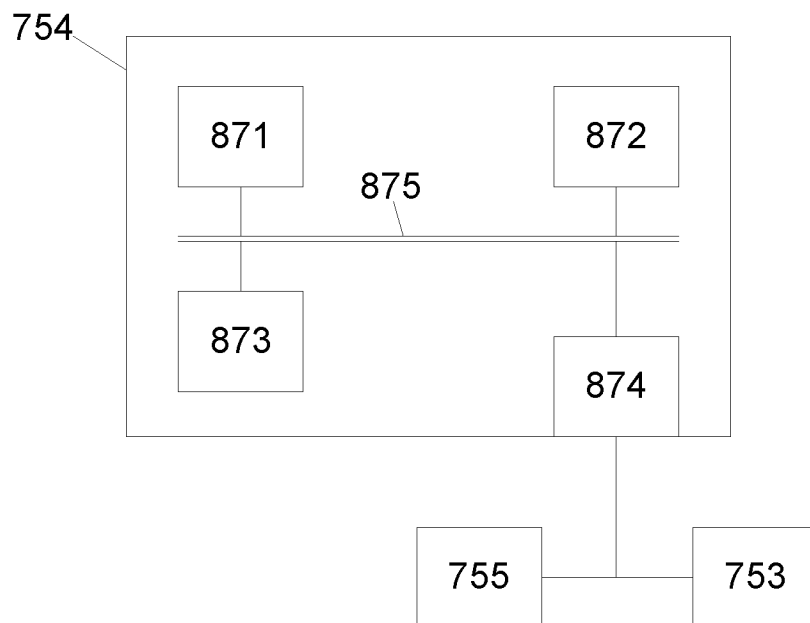


Fig. 8

Fig. 9A

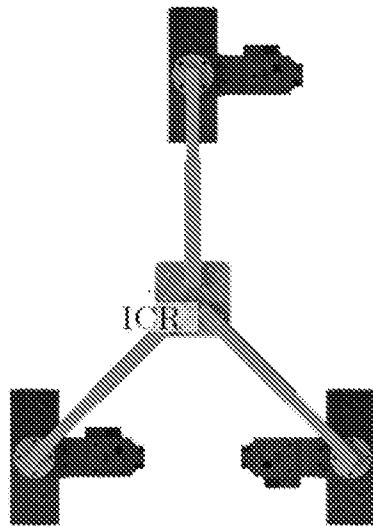


Fig. 9B

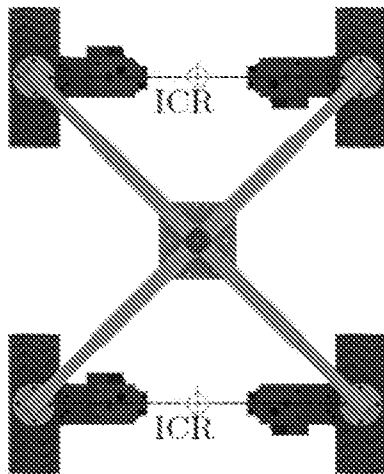
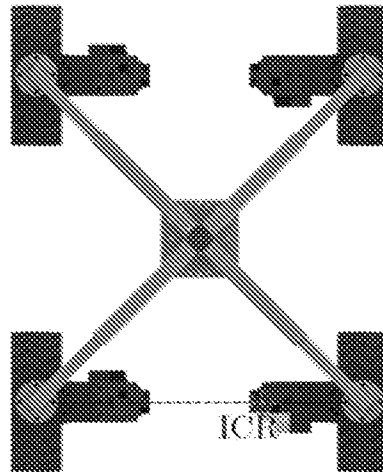


Fig. 9C

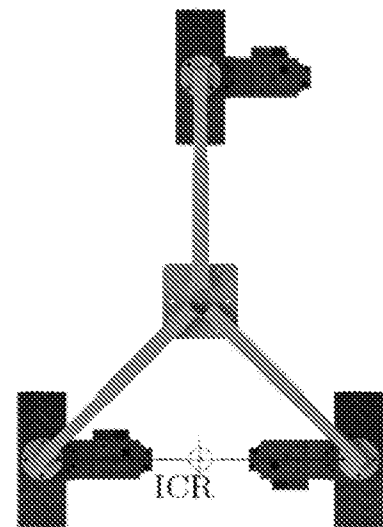
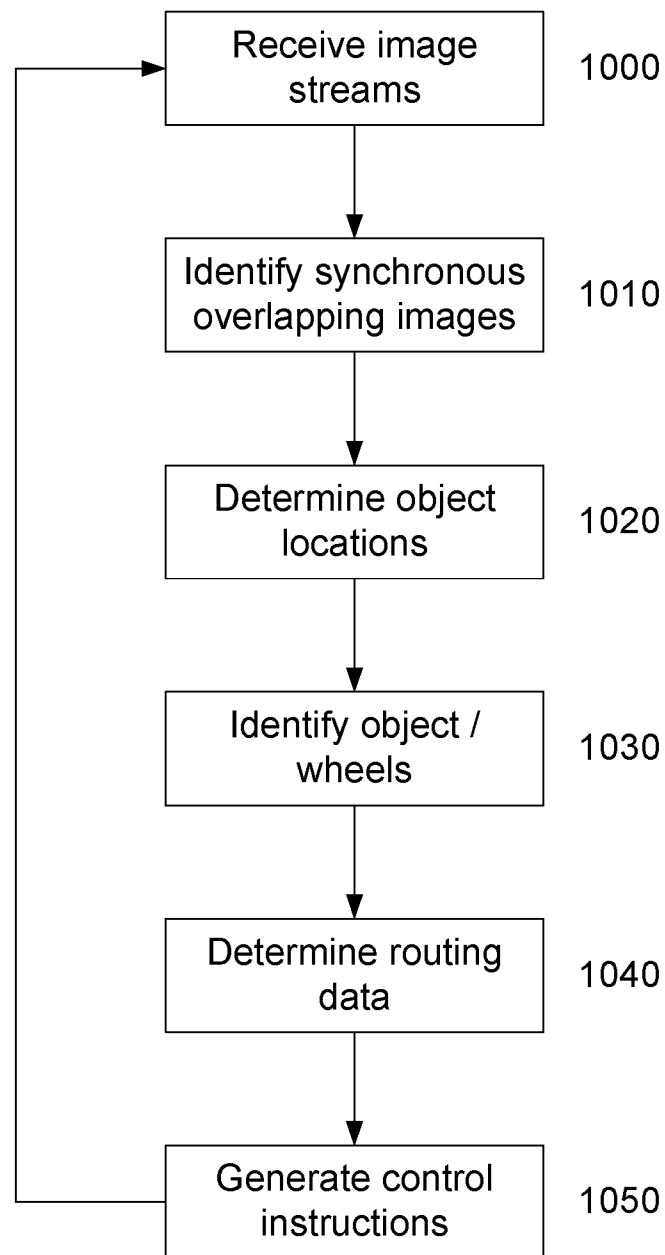


Fig. 9D

**Fig. 10**

9/19

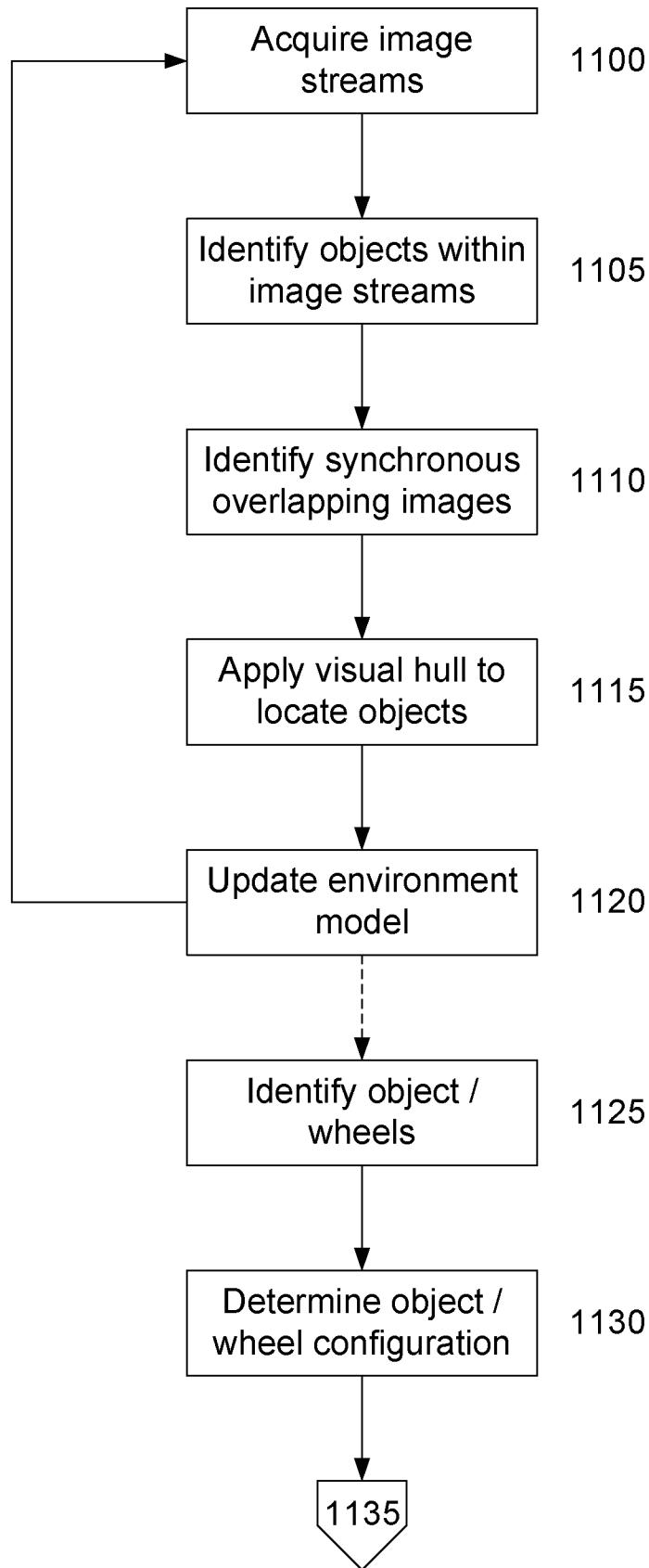


Fig. 11A

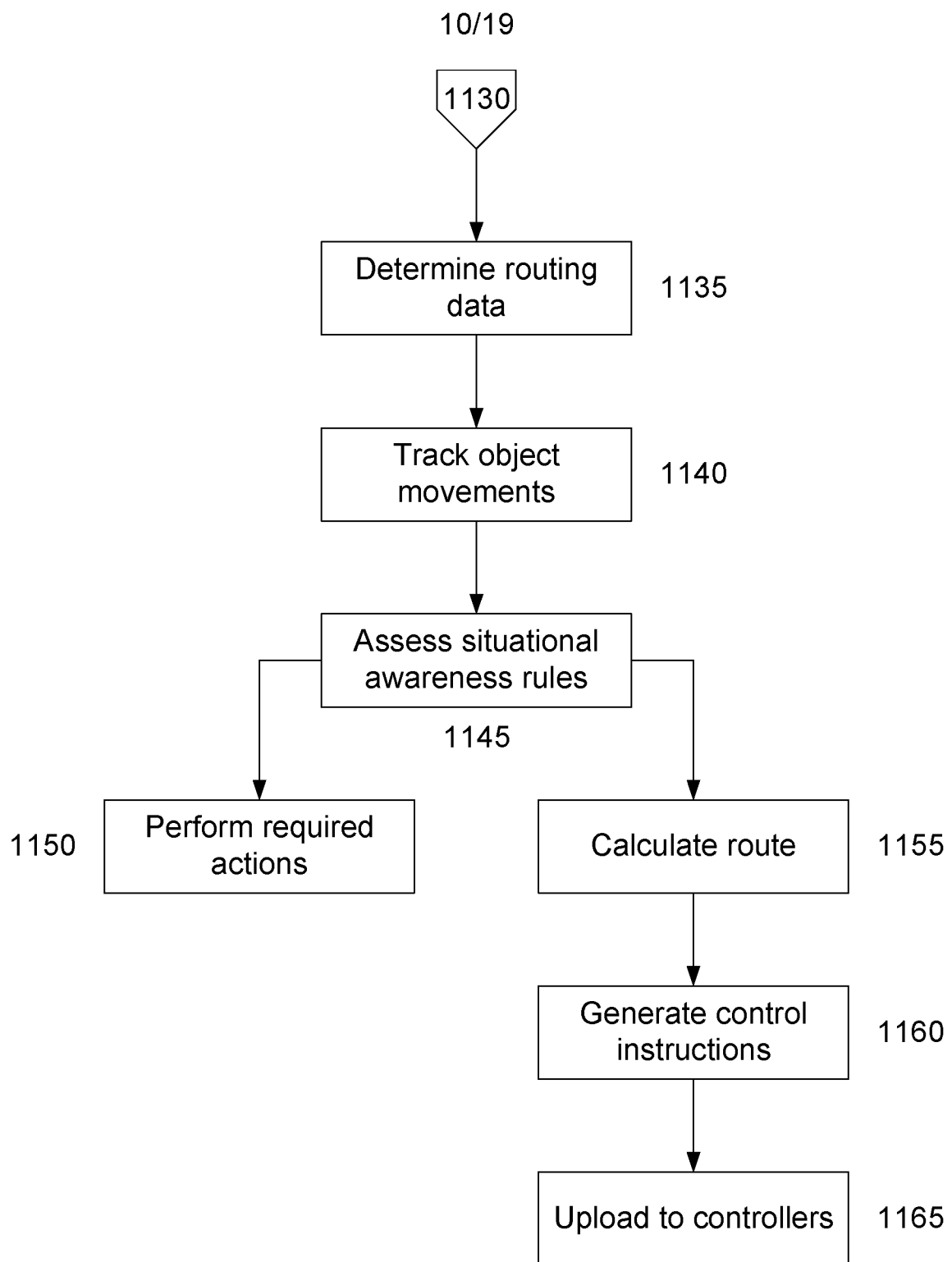


Fig. 11B

11/19

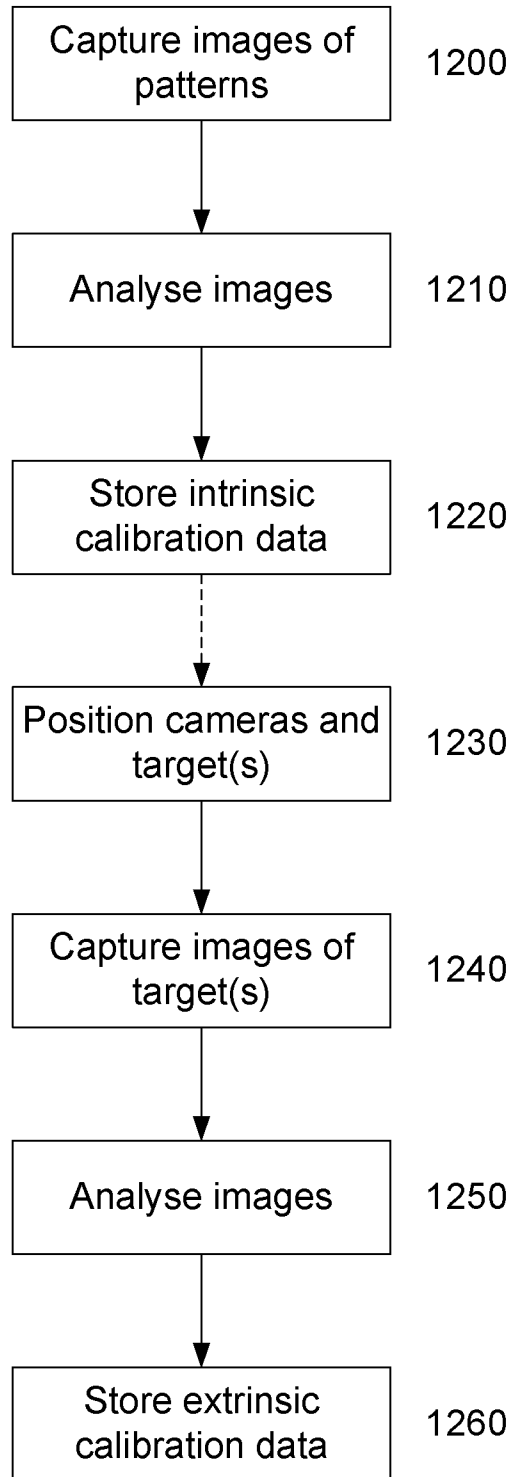


Fig. 12

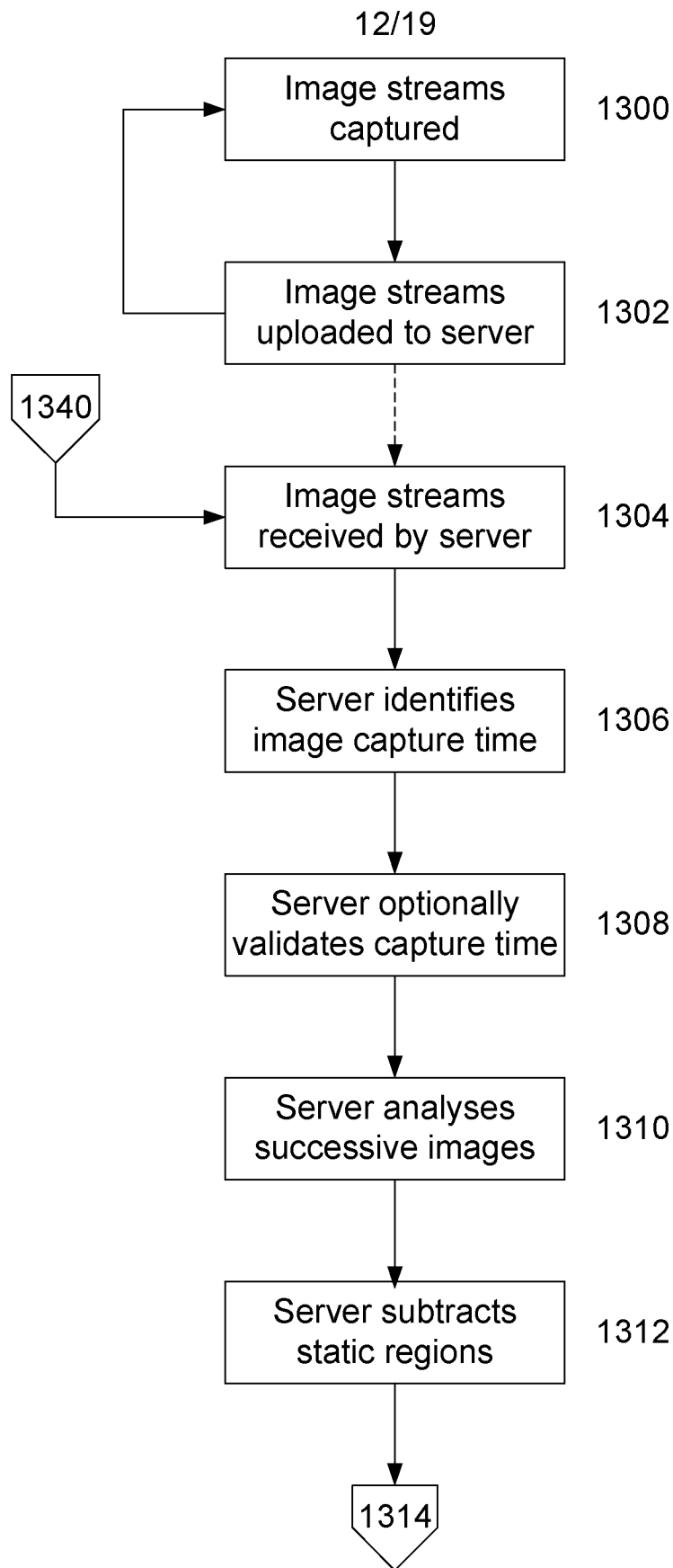


Fig. 13A

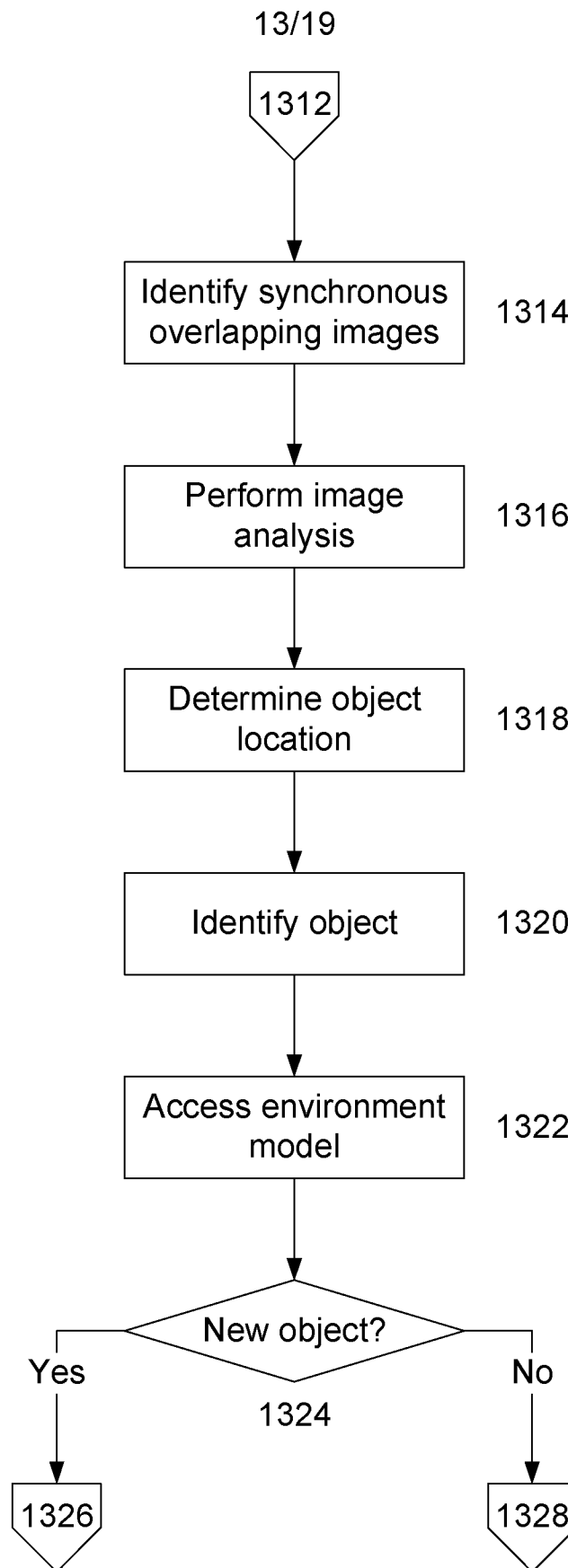
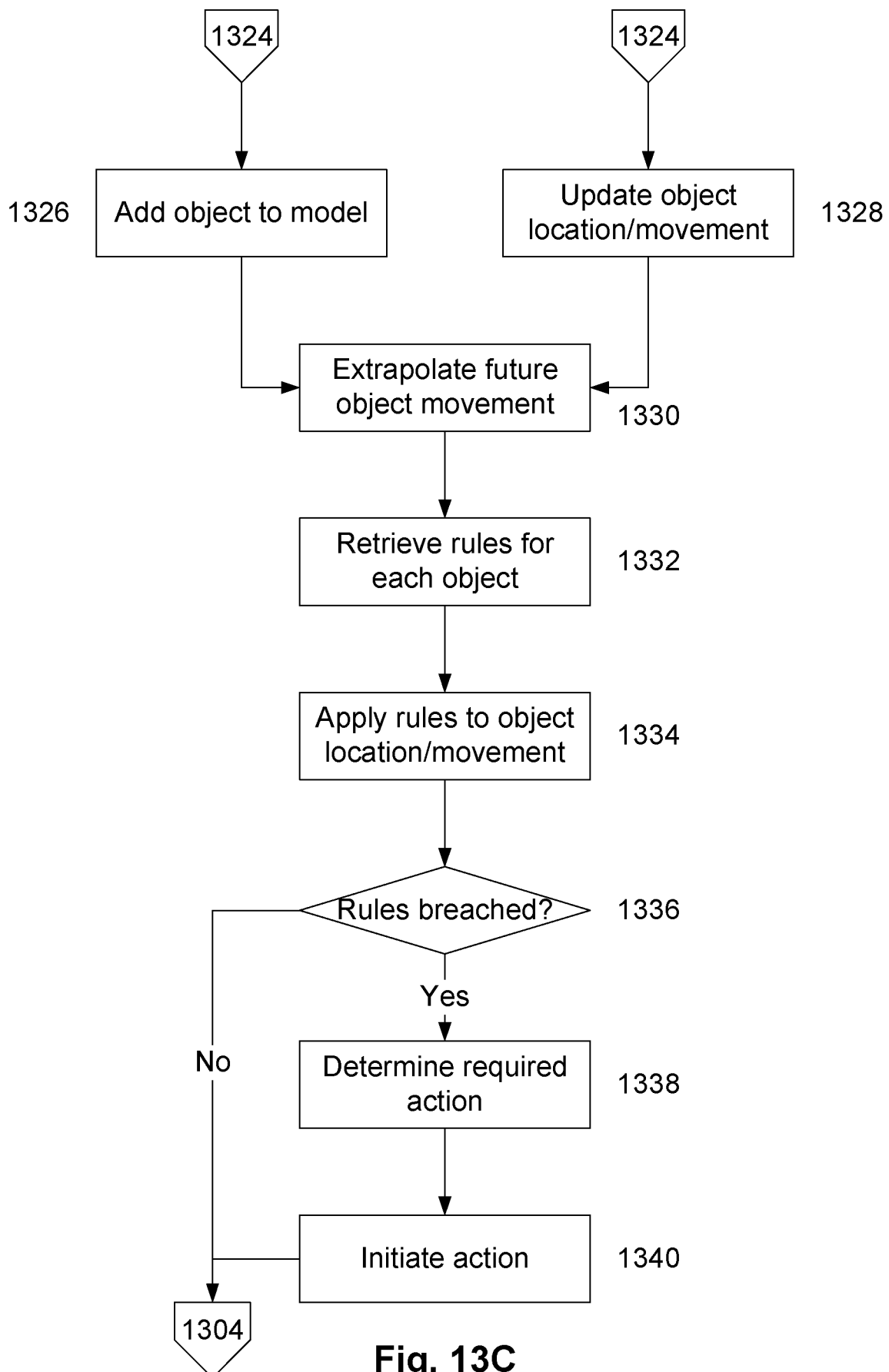
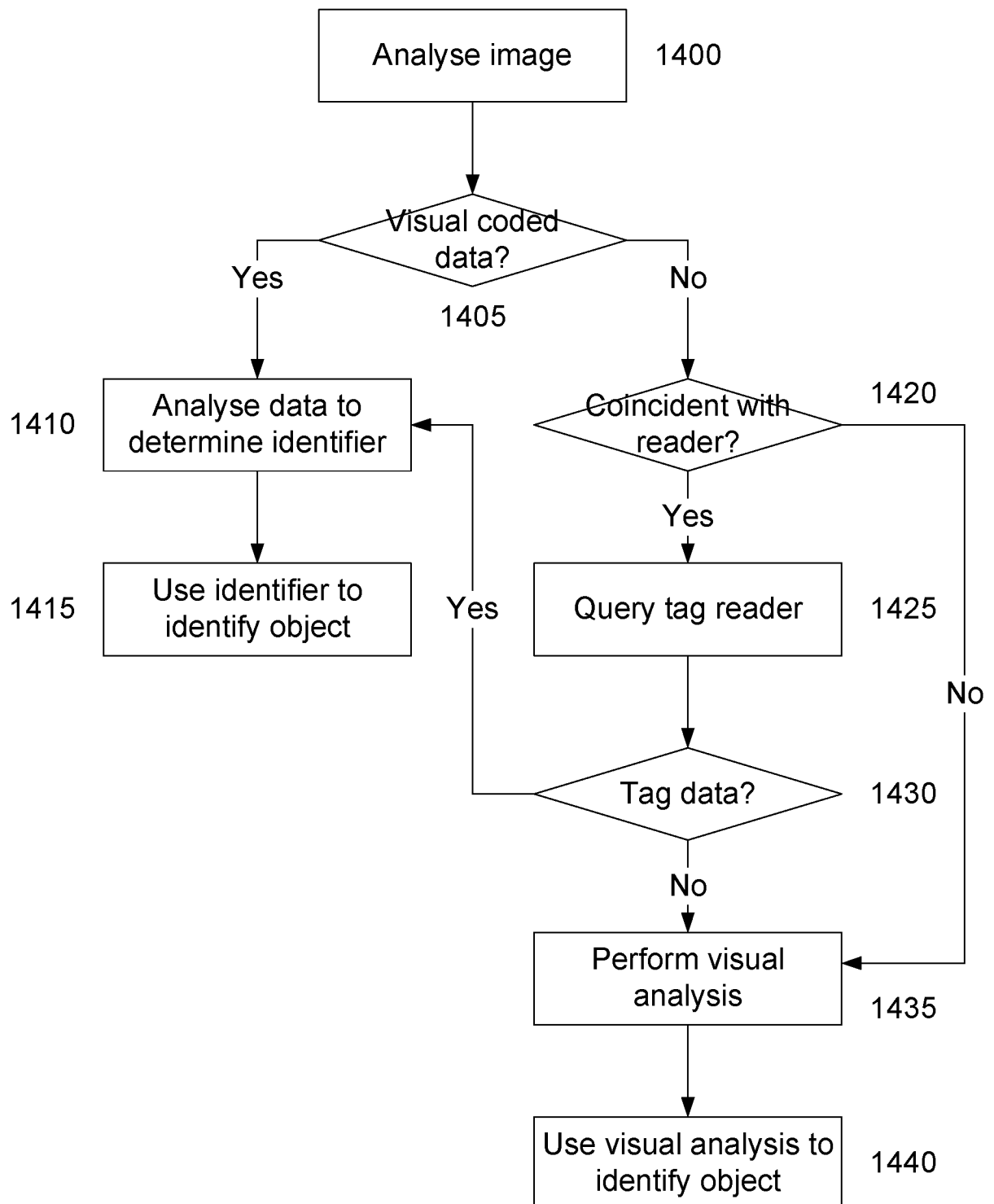


Fig. 13B

**Fig. 13C**

**Fig. 14**

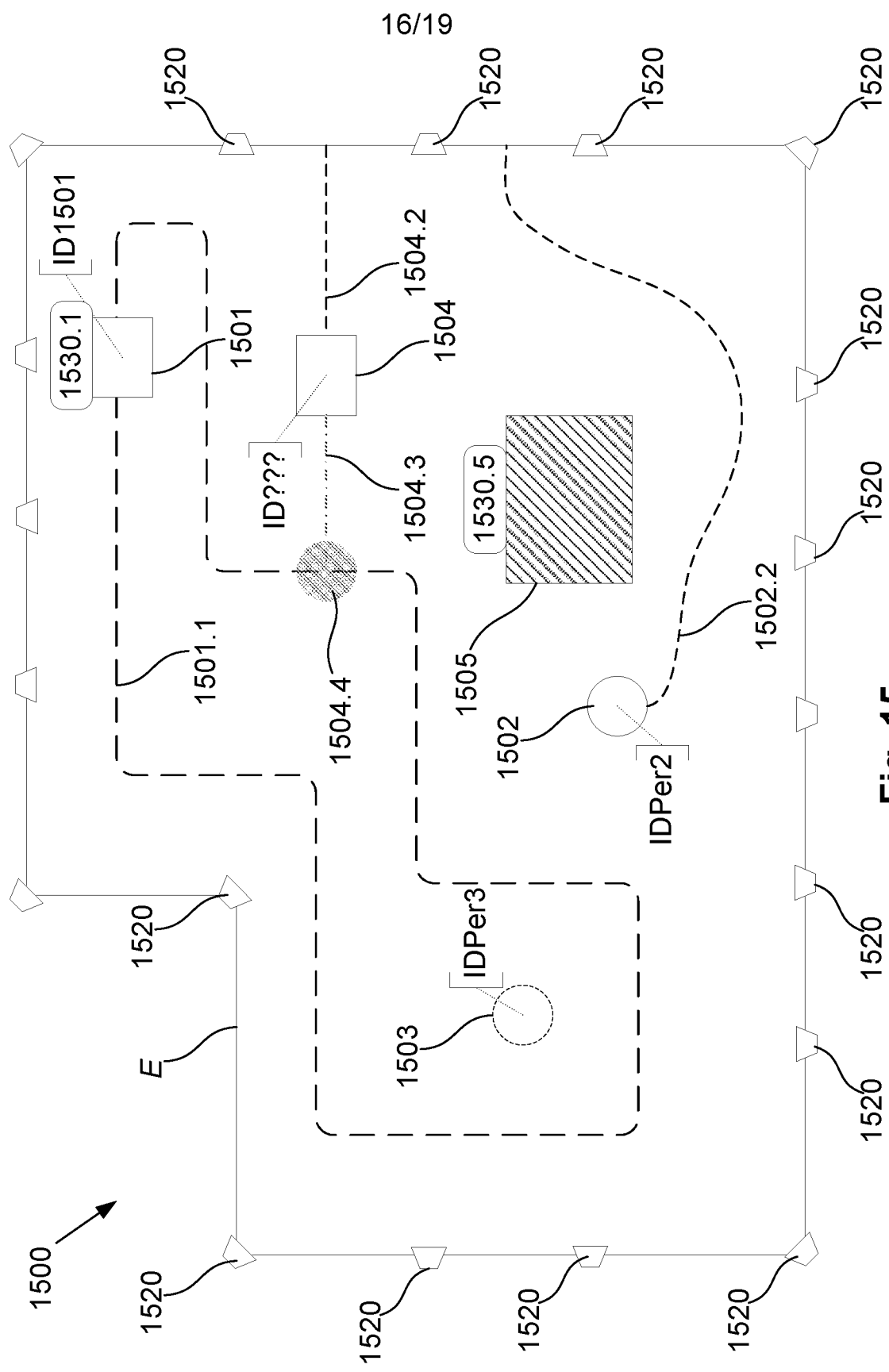


Fig. 15

17/19

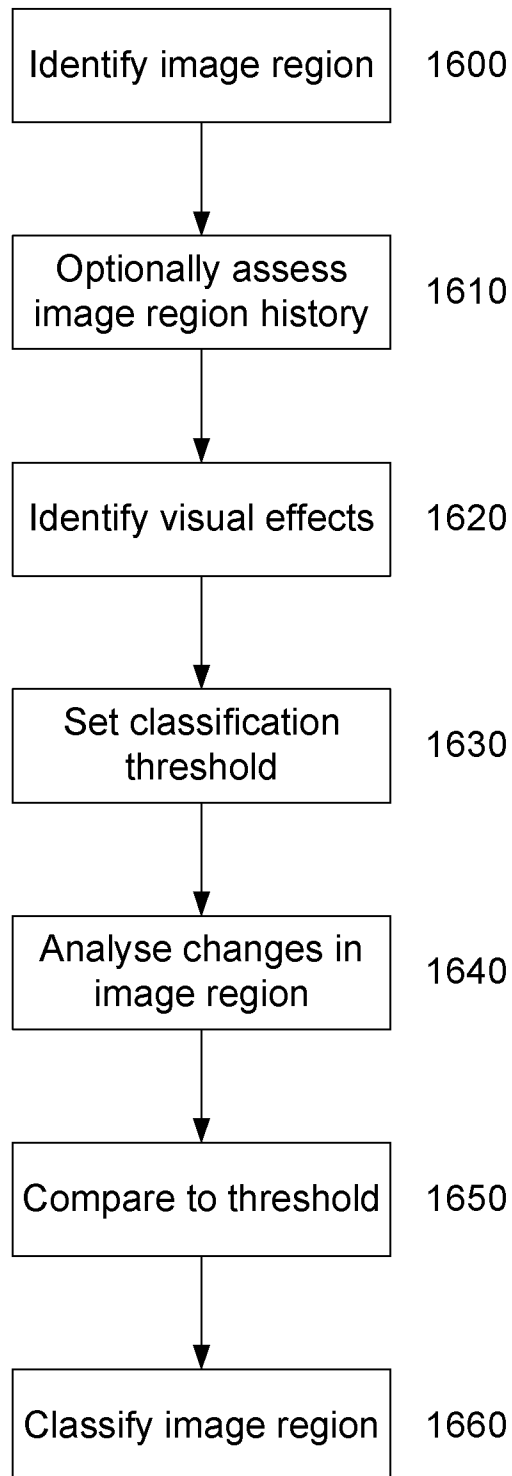


Fig. 16

18/19

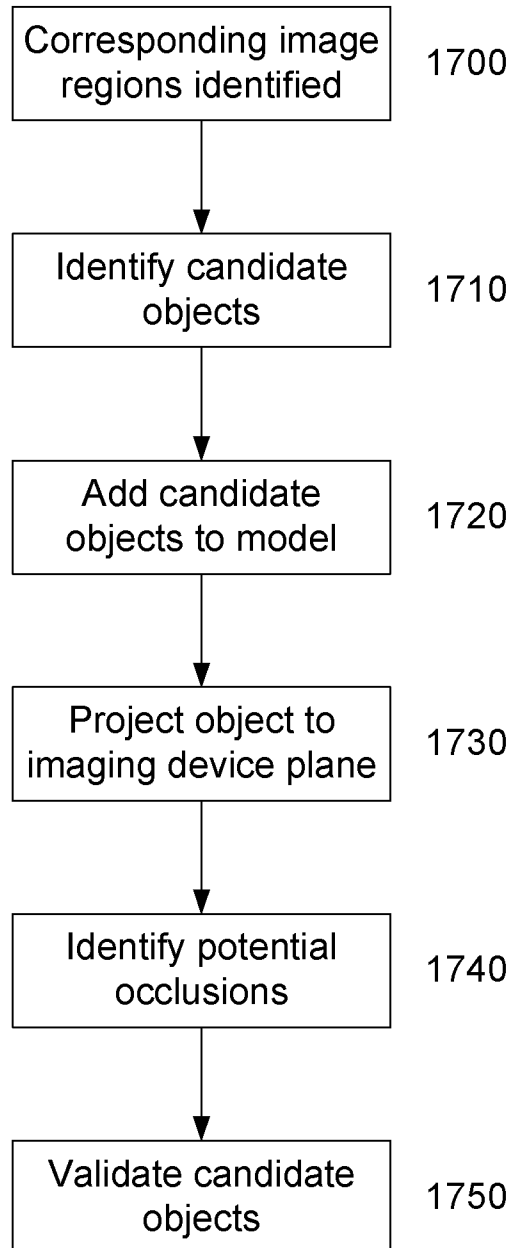


Fig. 17

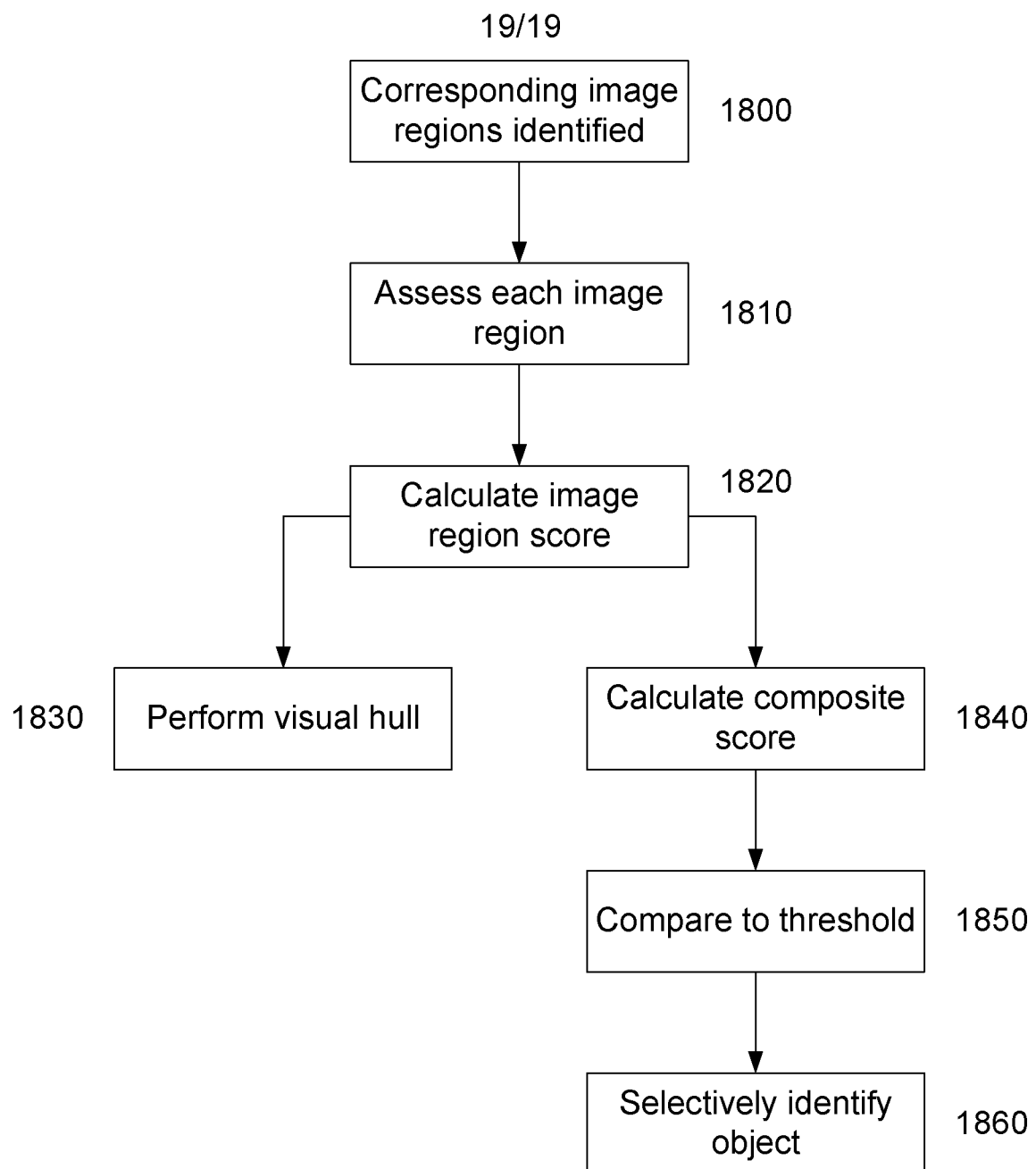


Fig. 18