



- (51) International Patent Classification: Not classified
- (21) International Application Number: PCT/US2012/041673
- (22) International Filing Date: 8 June 2012 (08.06.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 61/494,987 9 June 2011 (09.06.2011) US
- (71) Applicant (for all designated States except US): UTAH STATE UNIVERSITY RESEARCH FOUNDATION [US/US]; 1695 North Research Park Way, North Logan, Utah 84341 (US).
- (72) Inventor; and
- (71) Applicant : MATHIAS, Dean [US/US]; 2625 Black Walnut, North Logan, Utah 84341 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): DASU, Aravind [US/US]; 35 North Satsuma Dr., Providence, Utah 84332 (US). LIU, Chenguang [CN/US]; 10 Aggie Village, Apt. C, Logan, Utah 84341 (US). CHRISTENSEN, Randy [US/US]; 1761 E. High Creek Road, Cove, Utah 84320 (US). CHRISTENSEN, Bruce [US/US]; 1030 East 2100 North, North Logan, Utah 84341 (US).
- (74) Agent: RANDALL, Joshua N.; Holland & Hart LLP, P.O. Box 11583, 222 S. Main Street, Suite 2200, Salt Lake City, Utah 84110 (US).
- (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD,

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR SENSING OCCUPANCY

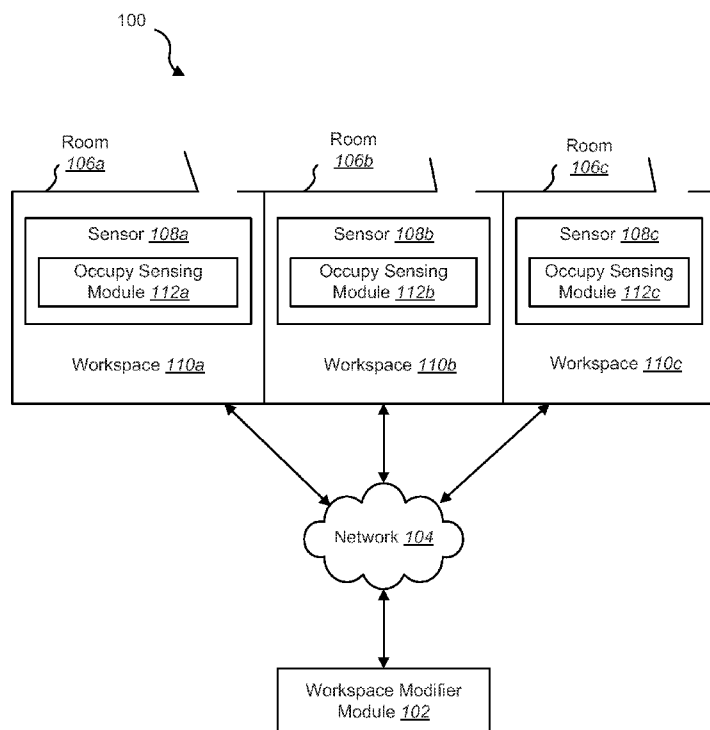


FIG. 1

(57) Abstract: A computer implemented method for sensing occupancy of a workspace includes creating a difference image that represents luminance differences of pixels in past and current images of the workspace resulting from motion in the workspace, determining motion occurring in regions of the workspace based on the difference image, and altering a workspace environment based at least in part on the determined motion. The method also includes determining which pixels in the difference image represent persistent motion that can be ignored and determining which pixels representing motion in the difference image are invalid because the pixels are isolated from other pixels representing motion.



SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, 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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))
- of inventorship (Rule 4.17(iv))

Published:

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

SYSTEMS AND METHODS FOR SENSING OCCUPANCY

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 61/494,987, filed June 9, 2012.

5

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH/DEVELOPMENT

[0002] This invention was made with government support under Contract No. DE-EE0003114 awarded by the U.S. Department of Energy. The government
10 has certain rights in the invention.

BACKGROUND

[0003] In various situations, motion and/or occupancy of individuals in a room may be detected for various reasons. For example, the lighting and/or climate
15 controls may be altered based on occupancy and/or the motion in the room. Altering the lighting, climate, etc. based on the motion and/or occupancy of a room by individuals may reduce energy costs.

DISCLOSURE OF THE INVENTION

20 [0004] A computer implemented method for sensing occupancy of a workspace includes creating a difference image that represents luminance differences of pixels between past and current images of the workspace resulting from motion in the workspace, determining motion occurring in regions of the workspace based on the difference image, and altering a workspace environment
25 based at least in part on the determined motion. The method also includes determining which pixels in the difference image represent persistent motion that can be ignored and determining which pixels representing motion in the difference image are invalid because the pixels are isolated from other pixels representing motion.

30 [0005] Another example of a computer implemented method for sensing occupancy of a workspace includes creating a difference image that represents

luminance differences of pixels in two sequential images of the workspace resulting from motion in the workspace, the difference image including motion pixels and non-motion pixels, determining which motion pixels are invalid by comparing the motion pixels to adjacent pixels and changing the motion pixels to non-motion pixels if the adjacent pixels are not also motion pixels, creating an updated difference image with the pixels changed from motion pixels to non-motion pixels, and altering a workspace environment based at least in part on the updated difference image.

[0006] A further computer implemented method for sensing occupancy of a workspace includes creating a difference image that represents luminance differences of pixels in two sequential images of the workspace resulting from motion in the workspace, the difference image including motion pixels and non-motion pixels, creating a persistence image having pixels corresponding to pixels of the difference image, and altering a workspace environment based at least in part on the persistence and difference images. A value of a pixel in the persistence image is increased each time a corresponding motion pixel is identified in the difference image, the value being decreased when a corresponding non-motion pixel is identified in the difference image, wherein the pixel in the persistence image is ignored when the value exceeds a threshold value.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The accompanying drawings illustrate a number of exemplary embodiments and are a part of the specification. Together with the following description, these drawings demonstrate and explain various principles of the instant disclosure.

[0008] FIG. 1 is a block diagram illustrating one embodiment of an environment in which the present systems and methods may be implemented.

[0009] FIG. 2 is a block diagram illustrating an example occupy sensing module in accordance with the present disclosure.

[0010] FIG. 3 is a block diagram illustrating one example of a occupy sensor in accordance with the present disclosure.

[0011] FIG. 4 is a schematic diagram of a digital image having a plurality of pixels.

[0012] FIG. 5 is a schematic diagram of an example difference image using the digital image of FIG. 4.

[0013] FIG. 6 is a schematic diagram of a corrected difference image.

[0014] FIG. 7 is a schematic diagram of an example persistence image
5 based on the corrected difference image of FIG. 6.

[0015] FIG. 8 is a schematic diagram of an example motion history image based on the corrected difference image of FIG. 7.

[0016] FIG. 9 is a block diagram illustrating an example room from the environment of FIG. 1.

10 [0017] FIG. 10 is a block diagram showing a relationship of states related to workspace occupancy according to the systems and methods described herein.

[0018] FIG. 11 is a flow diagram illustrating a portion of one example method of determining occupancy of a room.

[0019] FIG. 12 is a flow diagram illustrating another portion of the
15 example method of FIG. 11.

[0020] FIG. 13 is a flow diagram illustrating another portion of the example method of FIGS. 11 and 12.

[0021] FIG. 14 is a flow diagram illustrating another portion of the example method of FIGS. 11-13.

20 [0022] FIG. 15 is a flow diagram illustrating another portion of the example method of FIGS. 11-14.

[0023] FIG. 16 is a flow diagram illustrating an example method for sensing occupancy of a workspace.

[0024] FIG. 17 is a flow diagram illustrating another example method for
25 sensing occupancy of a workspace.

[0025] FIG. 18 is a flow diagram illustrating another example method for sensing occupancy of a workspace.

[0026] FIG. 19 depicts a block diagram of an electronic device suitable for implementing the present systems and methods.

30 [0027] While the embodiments described herein are susceptible to various modifications and alternative forms, specific embodiments have been shown by way of example in the drawings and will be described in detail herein. However, the

exemplary embodiments described herein are not intended to be limited to the particular forms disclosed. Rather, the instant disclosure covers all modifications, equivalents, and alternatives falling within the scope of the appended claims.

BEST MODE(S) FOR CARRYING OUT THE INVENTION

[0028] Building efficiency and energy conservation is becoming increasingly important in our society. One way to conserve energy is to power devices in a workspace only when those devices are needed. Many types of devices are needed only when the user is within a workspace or in close proximity to such devices. One scenario is an office workspace that includes a plurality of electronic devices such as lighting, heating and air conditioning, computers, telephones, etc. One aspect of the present disclosure relates to monitoring the presence of an occupant within the office workspace, and turning on and off at least some of the electronic devices based on the user's proximity to the office workspace.

[0029] An occupancy sensor system and related methods may be used to determine when an occupant's current location is within a given workspace. A sequence of images of the workspace may be used to determine the occupant's location. The sequence allows motion data to be extracted from the images. The current and past motion from the images comprises what may be referred to as motion history. Occupancy information may be used, for example, so that lighting within the space may be adjusted to maximally reduce energy consumption. Another example is altering room heating or cooling or providing other environmental controls in response to determining the occupant's location. Typically, the space to monitor and sense occupancy is referred to as a workspace, and typically has physical boundaries. For example, the workspace may have one or more fixed entrance locations that are monitored relative to other portions of the workspace.

[0030] Referring now to FIG. 1, an example system 100 is shown including a workspace modifier module 102 coupled via a network 104 to a plurality of occupy sensors 108a-c, which monitor workspaces 110a-c of rooms 106a-c, respectively. The sensors 108a-c monitor the workspaces 110a-c and include occupy sensing modules 112a-112c. The sensor modules 108a-c transmit data such as, for example,

occupied/unoccupied status of the rooms 106a-c to workspace modifier module 102 via a communications medium. The communications medium may include, for example, a network 104 such as a wireless or local area network.

[0031] The workspace modifier module 102 may be positioned remotely
5 from the rooms 106a-c. Alternatively, the workspace modifier module 102 may be housed locally in close proximity to one or more of the rooms 106a-c. Furthermore, the rooms 106a-c may be positioned adjacent to each other or be positioned at locations remote from each other. While a plurality of rooms 106a-c is shown in FIG. 1, other systems may include a single room or four or more rooms. Also, while
10 rooms 106a-c provide boundaries for the workspaces 110a-c, other arrangements are possible wherein the workspaces 110a-c are defined by other criteria including, for example, a sensing field of the sensors 108a-c.

[0032] Referring to FIG. 2, the occupy sensing module 112 may include a plurality of modules that perform various functions related to the systems and
15 methods. Occupy sensing module 112 may include, for example, a differencing sensing module 114, a noise reduction module 116, a motion persistence module 118, a motion history module 120, a motion evaluation module 122, a state machine update module 124, and an output module 126. At least some functions of the modules included in occupy sensing module 112 will be described below with
20 reference to some of the remaining figures. Various configurations may be possible for occupy sensing module 112 that include more or fewer modules than those shown in FIG. 2.

[0033] The differencing module 114 may perform comparison of past and current images and create the differencing image as described below with reference
25 to FIG. 5. The noise reduction module 116 may create updates to the differencing image as described below with reference to FIG. 6. The motion persistence module 118 may help identify persistent movement that can be ignored and create a persistence image as described below with reference to FIG. 7. The motion history module 120 may create a history of detected motion and a motion history image as
30 described below with reference to FIG. 8. The motion evaluation module 122 and state machine update module 124 may use the motion information from FIGS. 4-8 to

determine occupancy of a room as described below with reference to FIGS. 9 and 10 and the method of FIGS. 11-15.

[0034] The occupy sensors 108a include various components and modules in addition to an occupy sensing module 112, as shown in FIG. 3. For example, occupy sensor 108 may include an image generator 130, a transmitting module 132, and a settings module 134. The image generator 130 may include, for example, a camera and lens that collect digital images. The transmitting module 132 may provide communication between the occupy sensor and the workspace modifier module 102. The settings module 134 may provide adjustability of the settings for occupy sensor 108 including, for example, the timing in which images are collected and transmitted by occupy sensor 108.

[0035] Other types of sensors may be associated with the system 100 of FIG. 1. For example, motion sensors, heat sensors and the like may be used to help determine the presence of an occupant in the workspaces 110a-c.

[0036] Referring now to FIG. 4, a schematic digital image 180 is shown having a plurality of pixels labeled A1-An, B1-Bn, C1-Cn, D1-Dn, E1-E7 and F1-F2. The image 180 may include hundreds or thousands of pixels within the image. The image may be collected by the occupy sensor 108 (*e.g.*, the image generator 130). The image 180 may be delivered to the occupy sensing module 112 for further processing in accordance with the systems and methods disclosed herein.

[0037] Referring now to FIG 5, an example difference image 182 is shown with a plurality of pixels that correspond to the pixels of the image 180 shown in FIG. 4. The difference image 182 represents the difference between two sequential images 180 that are collected by the occupy sensor 108. The two sequential images may be referred to as a previous or prior image and a current image. For each pixel in the difference image 182, the absolute value of the difference in luminance between the current image and the previous image is compared to a threshold value. In at least one environment, if the difference is greater than the threshold value, the corresponding pixel in the difference image 182 is set to 1 or some other predefined value. If the difference is less than the threshold value, the corresponding pixel in the difference image is set to 0 or some other preset value. The color black may correspond to 0 and white may correspond to 1. The threshold value is chosen to be

an amount sufficient to ignore differences in luminance values that should be considered noise. The resulting difference image 182 contains a 1 (or white color) for all pixels that represent motion between the current image and the previous image and a 0 (or black color) for all pixels that represent no motion. The pixel C5
5 is identified in FIG. 5 for purposes of tracking through the example images described below with reference to FIGS. 6-8.

[0038] FIG. 6 shows a corrected difference image 184 that represents a correction to the difference image 182 wherein pixels representing motion in the difference image that should be considered invalid are changed because they are
10 isolated from other pixels in the image. Such pixels are sometimes referred to as snow and may be considered generally as “noise.” In one embodiment, each pixel in the difference image 182 that does not lie on the edge of the image and contains the value 1, retains the value of 1 if the immediate neighboring pixel (adjacent and diagonal) is also 1. Otherwise, the value is changed to 0. Likewise, each pixel with
15 a value of 0 may be changed to a value of 1 if the eight immediate neighboring pixels are 1, as shown for the pixel C5 in FIG. 6.

[0039] FIG. 7 schematically represents an example persistence image 186 that helps in determining which pixels in the corrected difference image 184 may represent persistent motion, which is motion that is typically considered a type of
20 noise and can be ignored. Each time a pixel in the corrected difference image 184 (or the difference image 182 if the correction shown in FIG. 6 is not made) represents valid motion, the value of the corresponding pixel in the persistence image 186 is incremented by 1. The incremental increase typically does not exceed a predetermined maximum value.

[0040] Each time a pixel in the corrected difference image 184 does not represent valid motion, the value of the corresponding pixel in the persistence image 186 is decremented. In one embodiment, the persistence image is decremented by 1, but may not go below 0. If the value of a pixel in a persistence image 186 is above a predetermined threshold, that pixel is considered to represent persistent motion.
25 Persistent motion is motion that reoccurs often enough that it should be ignored (e.g., a fan blowing in an office workspace). In the example of FIG. 7, if the
30

threshold value were 4, then the pixel C5 would have exceeded the threshold and the pixel C5 would represent persistent motion.

[0041] FIG. 8 schematically shows an example motion history image 188 that is used to help determine the history of motion in the workspace. In one embodiment, each time a pixel in the current image 180 represents valid, nonpersistent motion (*e.g.*, as determined using the corrected difference image 184 and the persistence image 186), the corresponding pixel in the motion history image 188 is set to a predetermined value such as, for example, 255. Each time a pixel in the current image 180 does not represent valid, nonpersistent motion, the corresponding pixel in the motion history image 188 is decremented by some predetermined value (*e.g.*, 1, 5, 20). This decremented value may be referred to as decay. The resulting value of each pixel in the motion history image 188 indicates how recently motion was detected in that pixel. The larger the value of a pixel in the motion history image 188, the more recent the motion occurred in that pixel.

[0042] FIG. 8 shows a value 255 in each of the pixels where valid, nonpersistent motion has occurred as determined using the corrected difference image 184 and the persistence image 186 (assuming none of the values in persistence image 186 have exceeded the threshold value). The pixels that had been determined as having either invalid or nonpersistent motion (a value of 0 in images 184, 186) have some value less than 255 in the motion history image 188.

[0043] Referring now to FIG. 9, a further step may be to evaluate the number of pixels that represent motion in particular regions in the image. Assuming the image 180 represents an entire footprint of the room 106a, the region in the image includes the area or regions defined by workspace 110a, border I 140, and border II 144. Borders may be next to or near an entrance to a workspace 110a. The number of pixels that represent valid, nonpersistent motion is counted for each of these regions. If there are no motion pixels in any of the regions, a motion disappeared trigger is enabled. If there are more motion pixels in the workspace 110a region than both of the border I 140 and border II 144 regions individually, a workspace motion trigger is enabled. If there are more motion pixels in border II 144 region than a predetermined threshold, a border II motion trigger is enabled. If there are more motion pixels in border I 140 region than border II 144 region, and if

there are more motion pixels in border I 140 region than the workspace 110a region, a border I motion trigger is enabled. If no motion has been detected for some predetermined timeout period, a safe timeout trigger is enabled.

[0044] The room 106a may include a further border reference as ignore border 146. In the event the sensor 108a is able to see through the entrance of the room 106a (e.g. through an open door) into a space beyond border I 140, movement beyond the ignore border 146 may be masked and ignored.

[0045] A state machine may be updated using the triggers generated in the region motion evaluation discussed with reference to FIG. 9. Each possible trigger may have an associated priority. In one embodiment, the decreasing priorities are: motion disappear, work space motion, board one motion, border two motion, and safe timeout. Each enabled trigger is evaluated in order of decreasing priority. If, for example, the currently evaluated trigger is the workspace motion trigger, the workspace motion signal updates the state machine and all other enabled triggers are discarded. This particular priority may be implemented because workspace motion makes any other motion irrelevant.

[0046] Proper placement in sizing of the borders shown in FIG. 9 and in other configurations may help optimize operation of the occupy sensing systems and methods discussed herein. Proper placement and size of the borders may allow the system to more accurately decide when an occupant has entered and departed a workspace. The borders may occupy enough pixels in the collected image such that the system may detect the occupant's presence within each of the borders. In one embodiment, three borders are used: border I 140, border II 144, and ignore border 146. Border I 140 is shown in FIG. 9 having a rectangular shape and may be positioned at the door opening. Border I 140 may be placed inside the workspace 110a and as near the doorway as possible without occupying any pixels that lie outside of the doorway within the ignore border 146. Typically, a side that is positioned adjacent to the door opening is at least as wide as the width of the door.

[0047] Border II 144 may be placed around at least a portion of the periphery of border I 140. Border II 144 may surround all peripheral surfaces of border I 140 that are otherwise exposed to the workspace 110a. Border II 144 may

be large enough that the system can detect the occupant's presence in border II 144 separate and distinct from detecting the occupant's presence in border I 140.

[0048] The ignore border 146 may also be rectangular in shape (or any other suitable shape) and is placed adjacent to border I 140 adjacent to the door opening. The ignore border 146 may be used to mask pixels in the image (*e.g.*, image 180) that are outside of the workspace 110a, but that are visible in the image. Any motion within the ignore border 146 is typically ignored.

[0049] A state machine may be used to help define the behavior of the occupy sensor and related system and methods. In one example, there are four states in the state machine: not occupy, border I motion, border II motion, and workspace occupy. Other examples may include more or fewer states depending on, for example, the number of borders established in the workspace. The not occupy state may be valid initially and when the occupant has moved from border I to somewhere outside of the workspace. If the occupant moves from border I to somewhere outside of the workspace, the workspace environment may be altered (*e.g.*, lights being turned off). The border I motion state may be valid when the occupant has moved into border I from either outside the workspace or from within the workspace. The border II motion state may be valid when the occupant has moved into border II from either border I or the workspace. If the occupant enters border II from border I, workspace environment may be altered (*e.g.*, the workspace lights are turned on). The workspace occupy state may be valid when the occupant has moved into the workspace from either border I or border II.

[0050] FIG. 10 schematically illustrates an example state machine having the four states described above. The state machine is typically set to not occupied 150 in an initial transition 174. Border I motion 152, border II motion 154, and workspace occupy state 156 are interconnected with arrows that represent the movement of the occupant from one space or border to another.

[0051] A motion disappear trigger may result, for example, in lights being turned off 158, and may occur as the occupant moves from border I 140 and into the ignore border 146. A border I motion trigger 160 may occur as the occupant moves from outside of the workspace 110a and into the border I 140. A border II motion trigger 162, resulting, for example, in turning a light on, may occur as the occupant

moves from border I 140 to border II 144. A border I motion trigger 164 may occur as the occupant moves from border II 144 to border I 140. A workspace motion trigger 166 may occur as the occupant moves from border II 144 to the workspace 110a. A border II motion trigger 168 may occur when an occupant moves from the workspace 110a to the border II 144. A workspace motion trigger 170 may occur as the occupant moves from border I 140 to the workspace 110a. A border I motion trigger 172 may occur as the occupant moves from the workspace 110a to border I 152.

[0052] Referring now to FIGS. 11-15, a detailed method 200 is described in which steps outlining a method for determining occupancy of a workspace from an image M pixels wide by N pixels high as shown. FIG. 11 shows the method 200 beginning with acquiring a first image 202 and initializing the count of pixels with motion to 0 in the step 204. A step 206 determines whether this is the first time through the method. If so, the method moves onto step 208 initializing an ignore region mask. If not, the system moves to step 216 and skips the steps of creating various data structures and the ignore mask region in steps 208, 210, 212 and 214.

[0053] Step 210 includes creating a data structure with dimensions M x N to store a binary difference image. Steps 212 includes creating a data structure with dimensions M x N to store the previous image. Step 214 includes creating a data structure with dimensions M x N to store a persistent motion image. The following step 216 includes copying a current image to the previous image data structure. For each pixel in the current image, if an absolute value of difference between the current pixel and corresponding pixel in a previous image is greater than a threshold, a corresponding value is set in a difference image to 1. Otherwise, a corresponding value is set in a difference image to 0 in the step 218. The step 220 includes leaving the value of a pixel at 1, for each pixel in the difference image set to 1, if the pixel is not on any edge of the image and all nearest neighbor pixels are set to 1. Otherwise, the pixel value is set at 0.

[0054] FIG. 12 shows further example steps of method 200. The method 200 may include determining for each pixel in the persistence image whether the corresponding pixel in the difference image is set to 1 in a step 222. Further step 224 includes incrementing the value of the pixel in the persistence image by 1, and

not to exceed a predefined maximum value. If the value of the corresponding pixel and the persistence image is greater than a predetermined threshold, the corresponding pixel is set in the motion history image to 0 in a step 226.

[0055] A step 226 includes determining whether a corresponding pixel in a difference image is set to 0. If so, step 230 includes decrementing a value of the corresponding pixel in a persistence image by 1, and not to decrease below the value of 0. If a corresponding pixel in the difference image is set to 1 and the condition in step 222 is yes and the condition in step 228 is no, then a further step 234 includes setting a value of the corresponding pixel in a motion history image 255 or some other predefined value. A step 236 includes increasing the dimension of the motion region rectangle to include this pixel. An increment count of pixels with motion is increased by 1 in the step 238.

[0056] FIG. 13 shows potential additional steps of method 200 including determining whether the condition in step 232 is no. If so, a step 242 includes decrementing a value of the corresponding pixel in the motion history image by a predetermined value, and not to decrease below a value of 0. If the value of the corresponding pixel in the motion history image is greater than 0, according to a step 244, a step 246 includes incrementing a count of pixels with motion by 1. A step 248 includes increasing a dimension of the motion region rectangle to include this pixel.

[0057] FIG. 14 shows potential additional steps of the method 200 including a step 250 of assigning a, b and c to be equal to the number of pixels from a motion region rectangle that lie within Border I 140, Border II 144, and the workspace 110a, respectively. If a, b and c are all 0, a motion disappear trigger is enabled in step 252. If c is greater than both a and b, a workspace motion trigger is enabled in a step 254. If b is greater than a predetermined threshold, a border II motion trigger is enabled in a step 256. If a is greater than b and a is greater than c, a border I motion is triggered in a step 258. If no motion is detected for a predetermined timeout period, a safe timeout trigger is enabled in a step 260. All enabled triggers may be added to a priority queue in a step 262. The priority may be arranged highest to lowest as according to a step 262: motion disappear, workspace motion, border I motion, border II motion, and safe timeout.

[0058] FIG. 15 shows further example steps of method 200. A step 264 includes updating a state machine based on the queue triggers. If a workspace motion trigger is in the priority queue, the trigger is removed from the queue and a workspace motion signal is issued to the state machine, according to a step 266. All of the other triggers may be removed from the priority queue. For each other trigger in the priority queue, a trigger with the highest priority is removed, a respective signal is issued to the state machine, and the trigger is removed from the queue according to a step 268. Further step 270 may include acquiring the next image and the process returns to step 204 according to step 272 through at least some of those steps shown and described with reference to method 200 of FIGS. 11-15.

[0059] Many other methods may be possible in accordance with the various systems, embodiments and features disclosed herein. An example method 300 is shown with reference to FIG. 16. A method 300 may include collecting past and current images of a workspace in a step 302. A step 304 may include creating the difference image that represents luminance differences in pixels in past and current images of the workspace resulting from motion in the workspace. A step 306 includes determining motion occurring in regions of the workspace based on the difference image. A step 308 includes altering a workspace environment based at least in part on the determined motion. Other steps of the method 300 may include determining which pixels in the difference image represent persistent motion that can be ignored in a step 310 and determining which pixels representing motion in the difference image are invalid because the pixels are isolated from other pixels representing motion in a step 312.

[0060] Another example step may include determining the number of pixels representing motion in each of a plurality region in the workspace and creating a trigger representative of which region has the most pixels representing motion. The method 300 may include evaluating the triggers based on a preset priority. The method 300 may include updating a state machine using a signal representing the evaluated triggers, wherein the state machine controls the workspace environment.

[0061] Creating the difference image according to method 300 may include comparing an absolute value of the difference in luminance between the images to a threshold value, setting a corresponding pixel in the difference image to 1 if the

difference is greater than the threshold value, and setting the corresponding pixel in the difference image to 0 if the difference is less than the threshold value. The method 300 may further include comparing every pixel representing motion to immediate neighboring pixels, and if the immediate neighboring pixels are not pixels
5 representing motion, changing a value of the pixel. The method 300 may include creating a persistence image having a pixel corresponding to each pixel in the difference image, wherein the pixels in the persistence image power increment in value each time a pixel in the difference image is determined to represent motion. The method 300 may include creating a motion history image representing how
10 recently motion was detected in each pixel representing motion in the difference image. A further step may include assigning a pixel value to pixels in the motion history image corresponding to pixels in the difference image representing motion, and subtracting from the pixel value each time a pixel in the current image is determined to not represent motion.

15 **[0062]** FIG. 17 shows another example method 400 that includes a step 402 of collecting past and current images of a workspace, and creating a difference image that represents luminance difference in pixels in two sequential images of the workspace resulting from motion in the workspace, wherein the difference image includes motion pixels and non-motion pixels in a step 404. A step 406 may include
20 determining which motion pixels are invalid by comparing the motion pixels to adjacent pixels and changing the motion pixels to non-motion pixels if the adjacent pixels are not also motion pixels. A step 408 may include creating an updated difference image with the pixels changed from motion pixels to non-motion pixels. A workspace environment may be altered based at least in part on the updated
25 difference image in step 410.

30 **[0063]** Another example step of method 400 may include comparing the motion pixels to a plurality of media adjacent of immediate neighboring pixels (*e.g.*, 8 immediate neighboring pixels). The method 400 may include determining the number of motion pixels in each of a plurality of regions in the workspace and creating a trigger representative of the region that has the most motion pixels. Method 400 may include updating a state machine using the triggers, wherein the state machine controls the workspace environment. Another step may include

creating a persistence image having a pixel corresponding to each pixel in a difference image, wherein the pixels in the persistence image are incremented in value each time a pixel on the difference image is determined to represent motion. The method 400 may include creating a motion history image representing how recently motion was detected in each motion pixel.

[0064] FIG. 18 shows another example method 500 that includes collecting past and current images of a workspace in step 502. The method 500 may include creating a difference image that represents luminance difference in pixels in two sequential images of the workspace resulting from motion in the workspace, wherein the difference image includes motion pixels and non-motion pixels in the step 504. Step 506 includes creating a persistence image having pixels corresponding to pixels of the difference image, the value of the pixels and the persistence image being increased each time a corresponding motion pixel is identified in the difference image, the value being decreased when a corresponding non-motion pixel is identified in the difference image, wherein the pixels in the persistence image are ignored when the value exceeds a threshold value. A step 508 includes altering a workspace environment based at least in part on the persistence and difference images.

[0065] The method 500 may also include determining a motion occurring in regions of the workspace based at least in part on the persistence and difference images. The method may also include creating a motion history image representing how recently a motion was detected in each motion pixel. The method 500 may include determining the number of pixels representing motion in each of the regions in the workspace and creating a trigger representative of the region having the most pixels representing motion. The method may include evaluating the triggers based on a pre-set priority. The method 500 may further include updating a state machine using a signal representing the trigger, the state machine controlling alterations to the workspace environment.

[0066] FIG. 19 depicts a block diagram of an electronic device 602 suitable for implementing the present systems and methods. The electronic device 602 includes a bus 610 which interconnects major subsystems of electronic device 602, such as a central processor 604, a system memory 606 (typically RAM, but which

may also include ROM, flash RAM, or the like), a communications interface 608, input devices 612, output device 614, and storage devices 616 (hard disk, floppy disk, optical disk, etc.).

[0067] Bus 610 allows data communication between central processor 604 and system memory 606, which may include read-only memory (ROM) or flash memory (neither shown), and random access memory (RAM) (not shown), as previously noted. The RAM is generally the main memory into which the operating system and application programs are loaded. The ROM or flash memory can contain, among other code, the Basic Input-Output system (BIOS) which controls basic hardware operation such as the interaction with peripheral components or devices. For example, the occupy sensing module 112 to implement the present systems and methods may be stored within the system memory 606. The occupy sensing module 112 may be an example of the occupy sensing module of FIGS. 1 and 2. Applications and/or algorithms resident with the electronic device 602 are generally stored on and accessed via a non-transitory computer readable medium (stored in the system memory 606, for example), such as a hard disk drive, an optical drive, a floppy disk unit, or other storage medium. Additionally, applications can be in the form of electronic signals modulated in accordance with the application and data communication technology when accessed via the communications interface 608

[0068] Communications interface 608 may provide a direct connection to a remote server or to the Internet via an internet service provider (ISP). Communications interface 608 may provide a direct connection to a remote server via a direct network link to the Internet via a POP (point of presence). Communications interface 608 may provide such connection using wireless techniques, including digital cellular telephone connection, Cellular Digital Packet Data (CDPD) connection, digital satellite data connection, or the like.

[0069] Many other devices or subsystems (not shown) may be connected in a similar manner. Conversely, all of the devices shown in FIG. 14 need not be present to practice the present systems and methods. The devices and subsystems can be interconnected in different ways from that shown in FIG. 14. The operation of an electronic device such as that shown in FIG. 14 is readily known in the art and is not discussed in detail in this application. Code to implement the present disclosure can

be stored in a non-transitory computer-readable medium such as one or more of system memory 606 and the storage devices 616.

[0070] Moreover, regarding the signals described herein, those skilled in the art will recognize that a signal can be directly transmitted from a first block to a second block, or a signal can be modified (*e.g.*, amplified, attenuated, delayed, latched, buffered, inverted, filtered, or otherwise modified) between the blocks. Although the signals of the above described embodiment are characterized as transmitted from one block to the next, other embodiments of the present systems and methods may include modified signals in place of such directly transmitted signals as long as the informational and/or functional aspect of the signal is transmitted between blocks. To some extent, a signal input at a second block can be conceptualized as a second signal derived from a first signal output from a first block due to physical limitations of the circuitry involved (*e.g.*, there will inevitably be some attenuation and delay). Therefore, as used herein, a second signal derived from a first signal includes the first signal or any modifications to the first signal, whether due to circuit limitations or due to passage through other circuit elements which do not change the informational and/or final functional aspect of the first signal.

[0071] While the foregoing disclosure sets forth various embodiments using specific block diagrams, flowcharts, and examples, each block diagram component, flowchart step, operation, and/or component described and/or illustrated herein may be implemented, individually and/or collectively, using a wide range of hardware, software, or firmware (or any combination thereof) configurations. In addition, any disclosure of components contained within other components should be considered exemplary in nature since many other architectures can be implemented to achieve the same functionality.

[0072] The process parameters and sequence of steps described and/or illustrated herein are given by way of example only and can be varied as desired. For example, while the steps illustrated and/or described herein may be shown or discussed in a particular order, these steps do not necessarily need to be performed in the order illustrated or discussed. The various exemplary methods described

and/or illustrated herein may also omit one or more of the steps described or illustrated herein or include additional steps in addition to those disclosed.

[0073] Furthermore, while various embodiments have been described and/or illustrated herein in the context of fully functional electronic devices, one or more of these exemplary embodiments may be distributed as a program product in a variety of forms, regardless of the particular type of computer-readable media used to actually carry out the distribution. The embodiments disclosed herein may also be implemented using software modules that perform certain tasks. These software modules may include script, batch, or other executable files that may be stored on a computer-readable storage medium or in an electronic device. In some embodiments, these software modules may configure an electronic device to perform one or more of the exemplary embodiments disclosed herein.

[0074] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the present systems and methods and their practical applications, to thereby enable others skilled in the art to best utilize the present systems and methods and various embodiments with various modifications as may be suited to the particular use contemplated.

[0075] Unless otherwise noted, the terms “a” or “an,” as used in the specification and claims, are to be construed as meaning “at least one of.” In addition, for ease of use, the words “including” and “having,” as used in the specification and claims, are interchangeable with and have the same meaning as the word “comprising.”

WHAT IS CLAIMED IS:

1. A computer implemented method for sensing occupancy of a workspace, comprising:

creating a difference image that represents luminance differences of pixels
5 between past and current images of the workspace resulting from motion in the workspace;

determining motion occurring in regions of the workspace based on the difference image;

altering a workspace environment based at least in part on the determined motion;
10

determining which pixels in the difference image represent persistent motion that can be ignored; and

determining which pixels representing motion in the difference image are invalid because the pixels are isolated from other pixels representing motion.
15

2. The method of claim 1, further comprising:

determining the number of pixels representing motion in each of a plurality of regions in the workspace and creating a trigger representative of the region having the most pixels representing motion.
20

3. The method of claim 2, further comprising:

evaluating the triggers based on a pre-set priority.

4. The method of claim 1, further comprising:
25

updating a state machine using a signal representing the evaluated triggers, the state machine controlling the workspace environment.

5. The method of claim 1, wherein creating the difference image includes comparing an absolute value of the difference in luminance between the images to a threshold value, setting a corresponding pixel in the difference image to 1 if the difference is greater than the threshold value, and setting the corresponding pixel in the difference image to 0 if the difference is less than the threshold value.

6. The method of claim 1, further comprising:
comparing every pixel representing motion to immediate neighboring pixels,
and if the immediate neighboring pixels are not pixels representing
motion, changing a value of the pixel.

7. The method of claim 1, further comprising:
creating a persistence image having a pixel corresponding to each pixel in the
difference image, the pixels in the persistence image being incremented
in value each time a pixel in the difference image is determined to
represent motion.

8. The method of claim 1, further comprising:
creating a motion history image representing how recently motion was
detected in each pixel representing motion in the difference image.

9. The method of claim 8, further comprising:
assigning a pixel value to pixels in the motion history image corresponding to
pixels in the difference image representing motion, and subtracting
from the pixel value each time a pixel in the current image is
determined to not represent motion.

10. A computer implemented method for sensing occupancy of a workspace, comprising:

creating a difference image that represents luminance differences of pixels in two sequential images of the workspace resulting from motion in the workspace, the difference image including motion pixels and non-motion pixels;

determining which motion pixels are invalid by comparing the motion pixels to adjacent pixels and changing the motion pixels to non-motion pixels if the adjacent pixels are not also motion pixels;

creating an updated difference image with the pixels changed from motion pixels to non-motion pixels; and

altering a workspace environment based at least in part on the updated difference image.

11. The method of claim 10, further comprising:

determining the number of motion pixels in each of a plurality of regions in the workspace, and creating a trigger representative of the region which has the most motion pixels.

12. The method of claim 10, further comprising:

updating a state machine using the triggers, the state machine controlling the workspace environment.

13. The method of claim 10, further comprising:

creating a persistence image having a pixel corresponding to each pixel in the difference image, the pixels in the persistence image being incremented in value each time a pixel in the difference image is determined to represent motion.

14. The method of claim 12, further comprising:
creating a motion history image representing how recently motion was
detected in each motion pixel.

5 15. A computer implemented method for sensing occupancy of a workspace,
comprising:

creating a difference image that represents luminance differences of pixels in
two sequential images of the workspace resulting from motion in the
workspace, the difference image including motion pixels and non-
10 motion pixels;

creating a persistence image having pixels corresponding to pixels of the
difference image, a value of the pixels in the persistence image being
increased each time a corresponding motion pixel is identified in the
difference image, the value being decreased when a corresponding non-
15 motion pixel is identified in the difference image, wherein the pixels in
the persistence image are ignored when the value exceeds a threshold
value; and

altering a workspace environment based at least in part on the persistence and
difference images.

20 16. The method of claim 15, further comprising:
determining motion occurring in regions of the workspace based at least in
part on the persistence and difference images.

25 17. The method of claim 15, further comprising:
creating a motion history image representing how recently motion was
detected in each motion pixel.

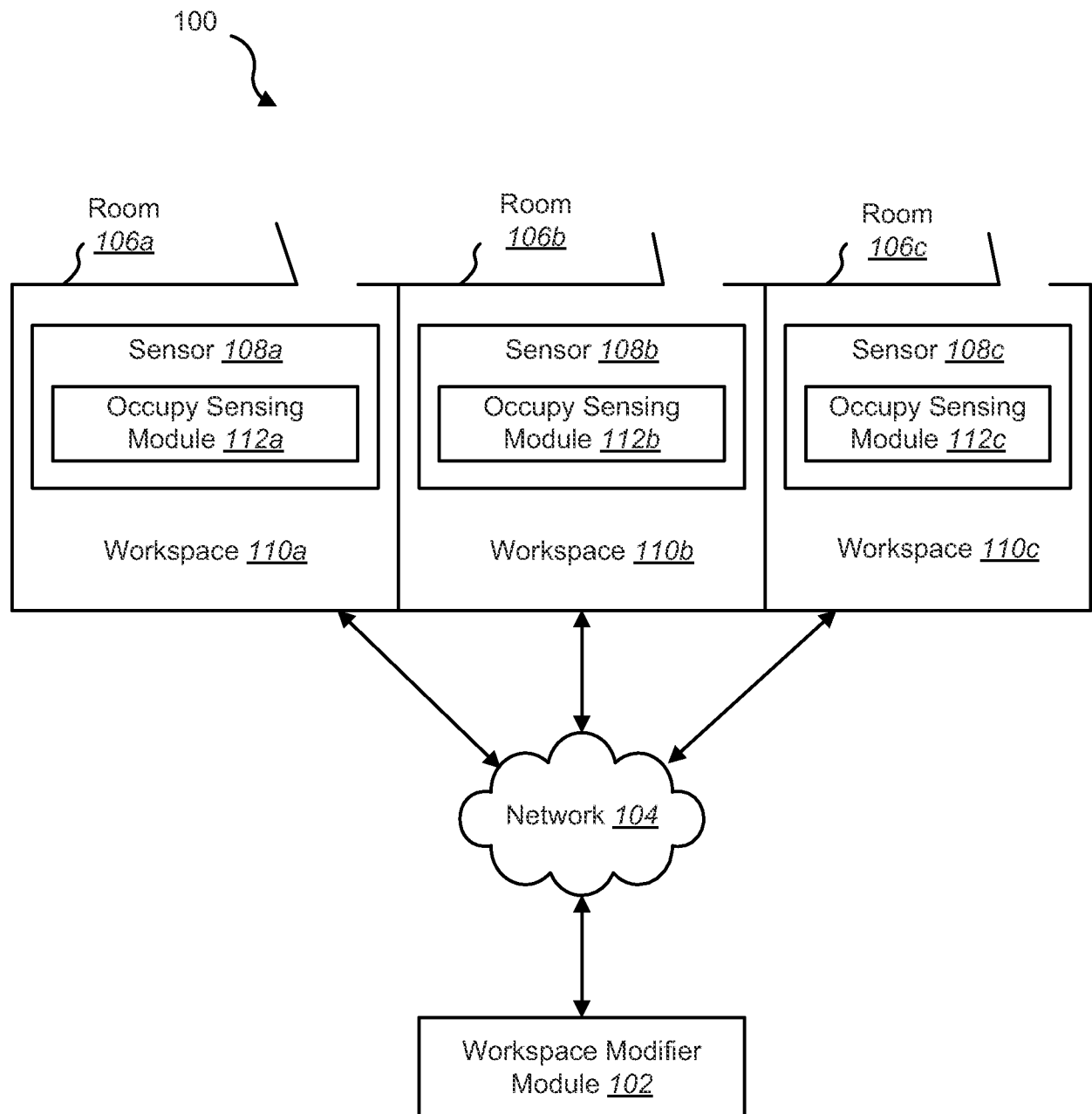
18. The method of claim 16, further comprising:
determining the number of pixels representing motion in each of the regions
in the workspace and creating a trigger representative of the region
having the most pixels representing motion.

5

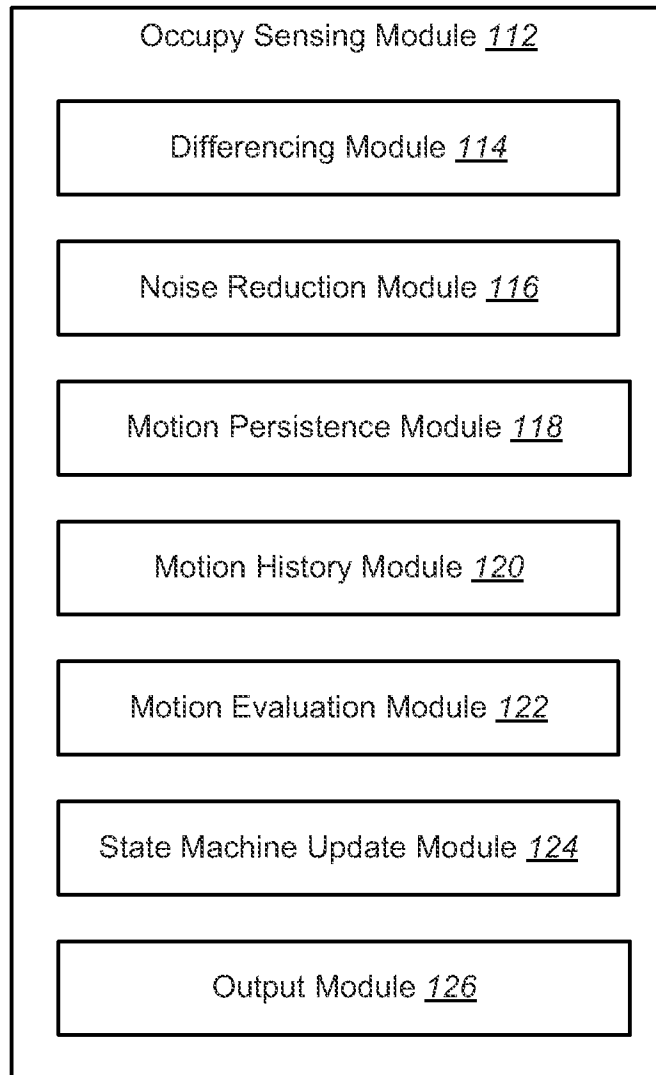
19. The method of claim 18, further comprising:
evaluating the trigger based on a pre-set priority.

20. The method of claim 18, further comprising:
10 updating a state machine using a signal representing the trigger, the state
machine controlling alterations to the workspace environment.

1/19

**FIG. 1**

2/19

**FIG. 2**

3/19

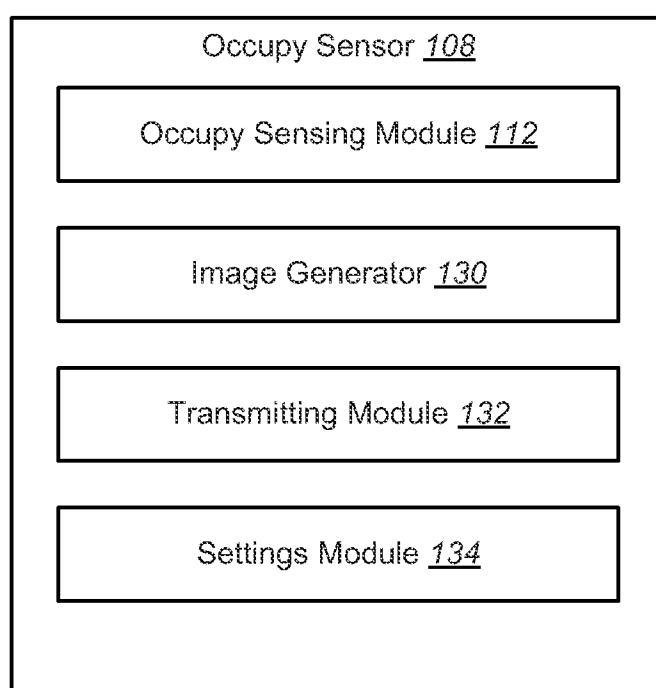


FIG. 3

180



A1	A2	A3	A4	A5	A6	A7	...	An
B1	B2	B3	B4	B5	B6	B7	...	Bn
C1	C2	C3	C4	C5	C6	C7	...	Cn
D1	D2	D3	D4	D5	D6	D7	...	Dn
E1	E2	E3	E4	E5	E6	E7		
F1	F2							

FIG. 4

184

0	0	0	0	1	1	1	1	...	0
0	0	0	0	1	1	1	1	...	0
0	0	1	1	1	1	0	0	...	0
0	0	0	1	1	1	0	0	...	0
0	0	0	0	1	1	1	1		
0	0	0	0	1	1	1	1		
0	0	0	0	0	0	0	0		
0	0	0	0						

C5

FIG. 6

186

0	0	0	1	1	1	1	...	0
0	0	0	2	2	3	1	...	0
0	0	1	3	5	3	0	...	0
0	0	0	1	2	2	0	...	0
0	0	0	1	1	1	1		
0	0	0						
0	0	0						

C5

FIG. 7

188

56	100	160	255	255	255	255	255	...	12
140	155	200	255	255	255	255	255	...	10
135	158	255	255	255	255	255	231	...	12
98	165	168	255	255	255	255	210	...	15
78	110	135	255	255	255	255	255		
65	101								

C5

FIG. 8

9/19

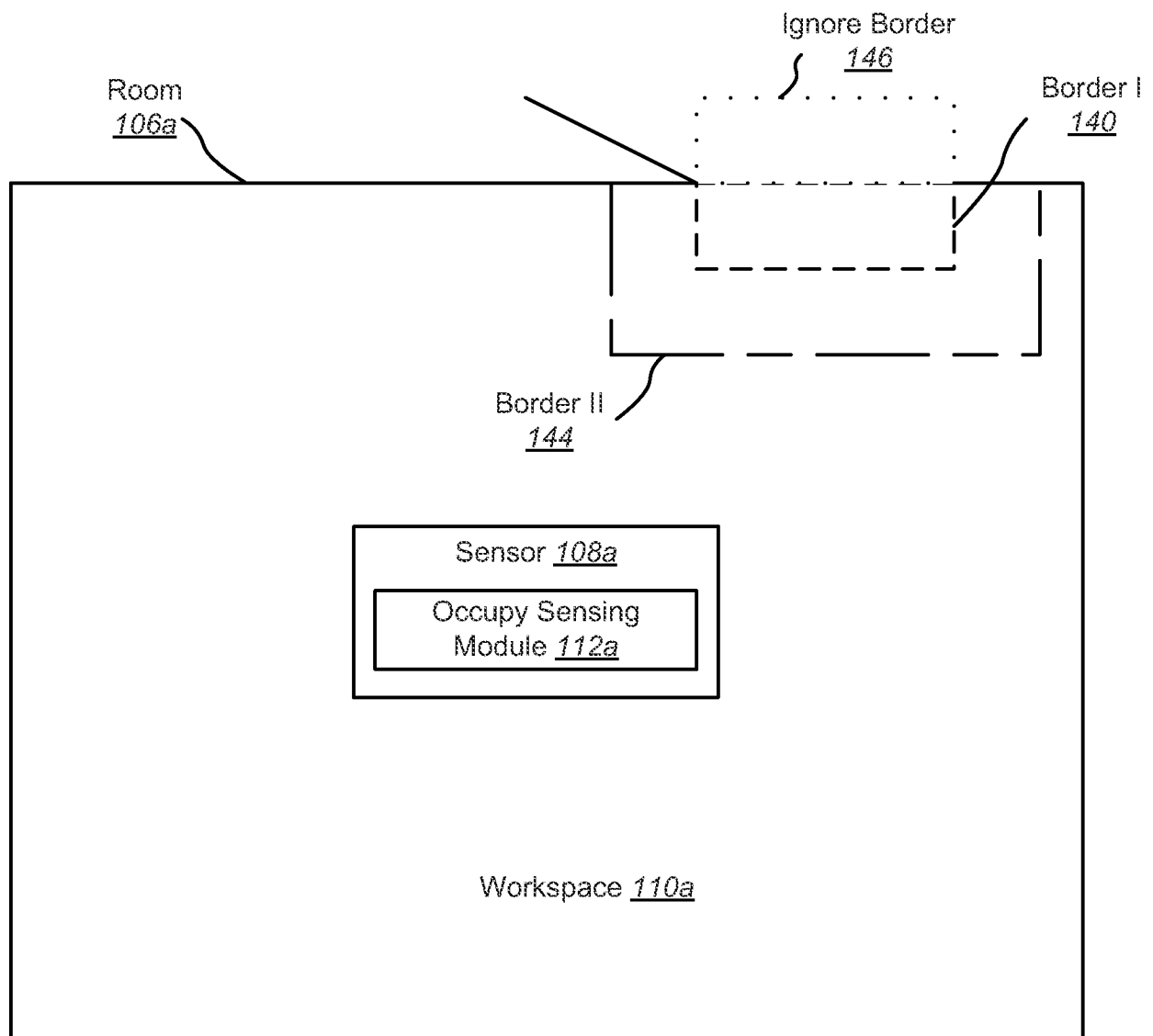
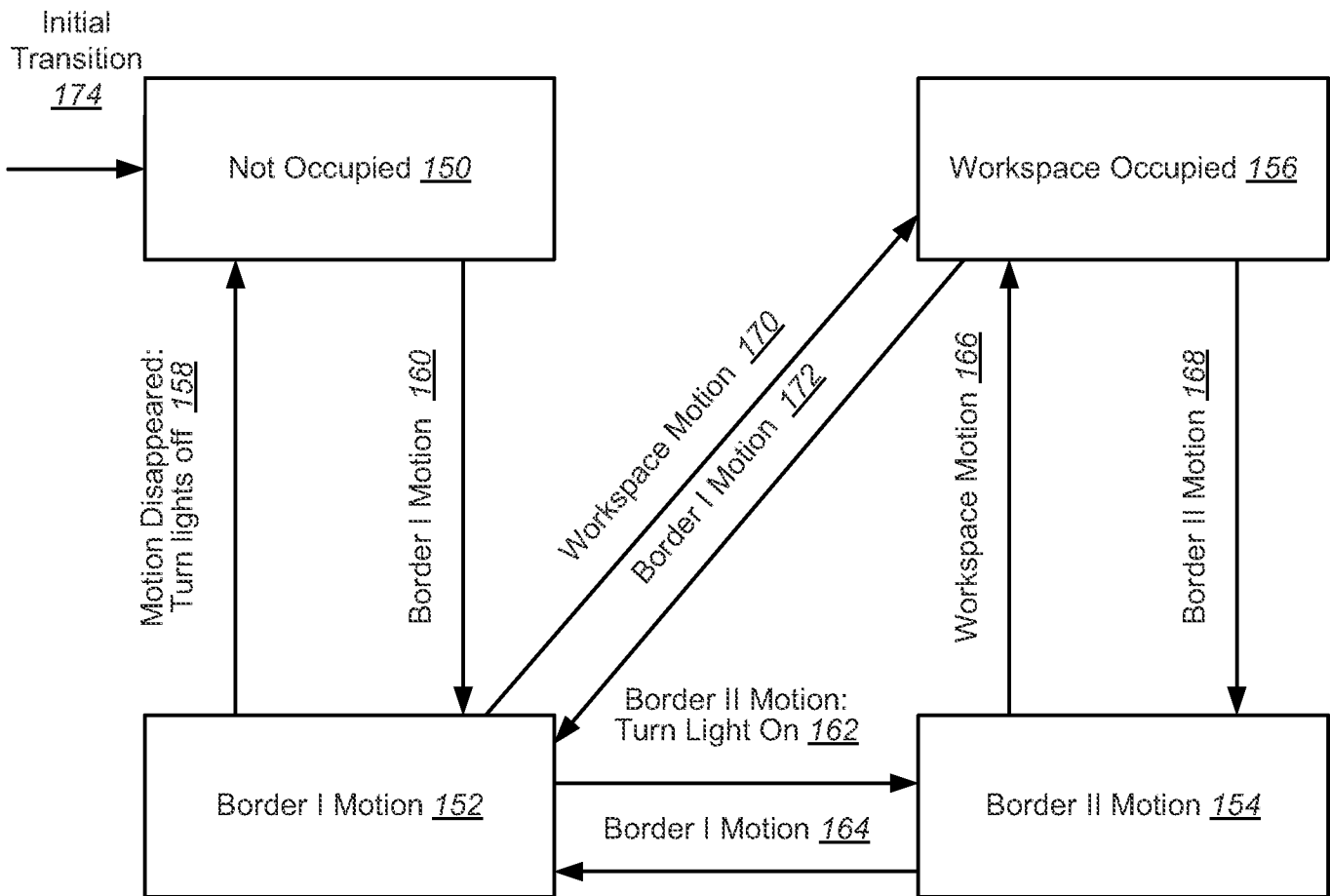
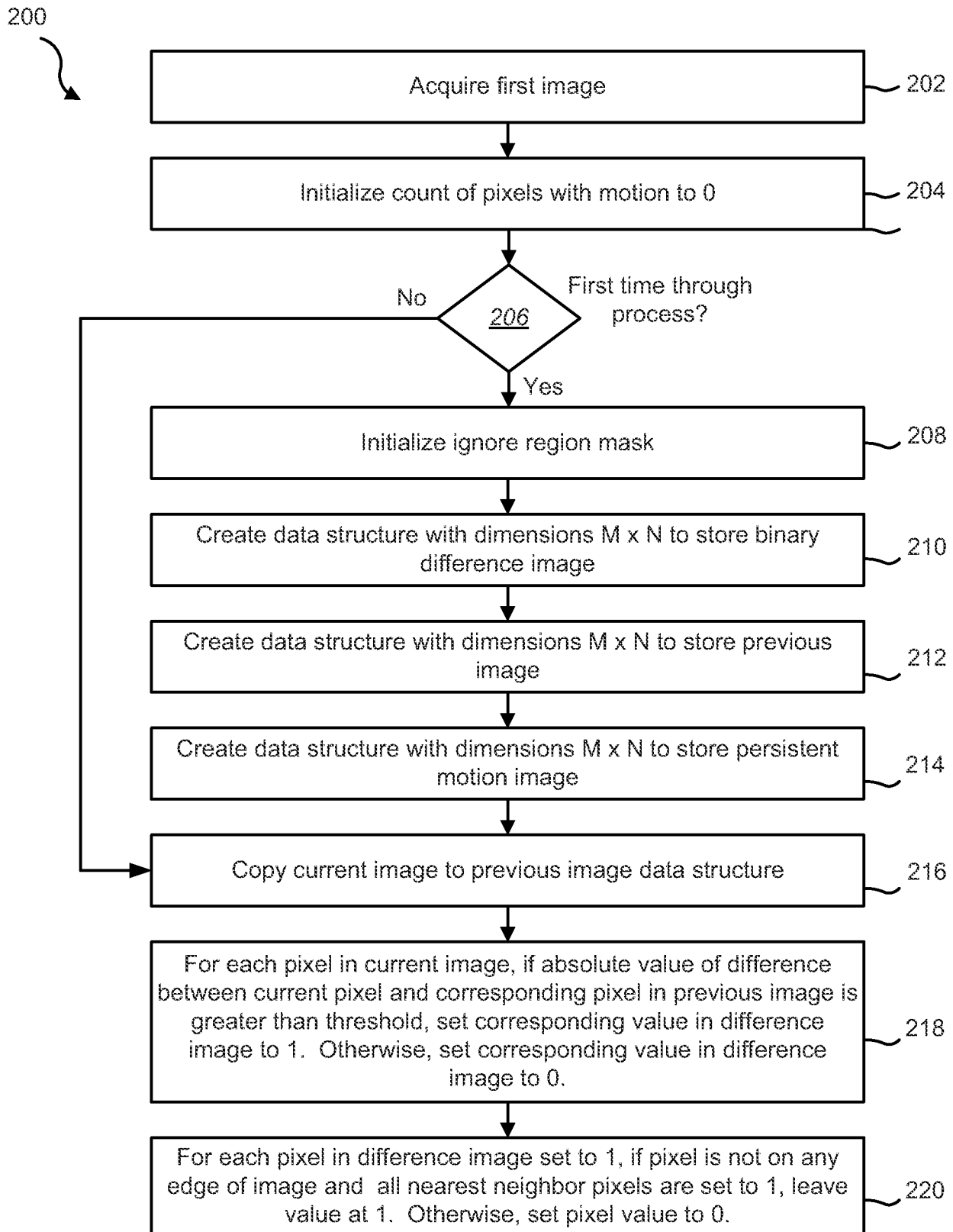


FIG. 9

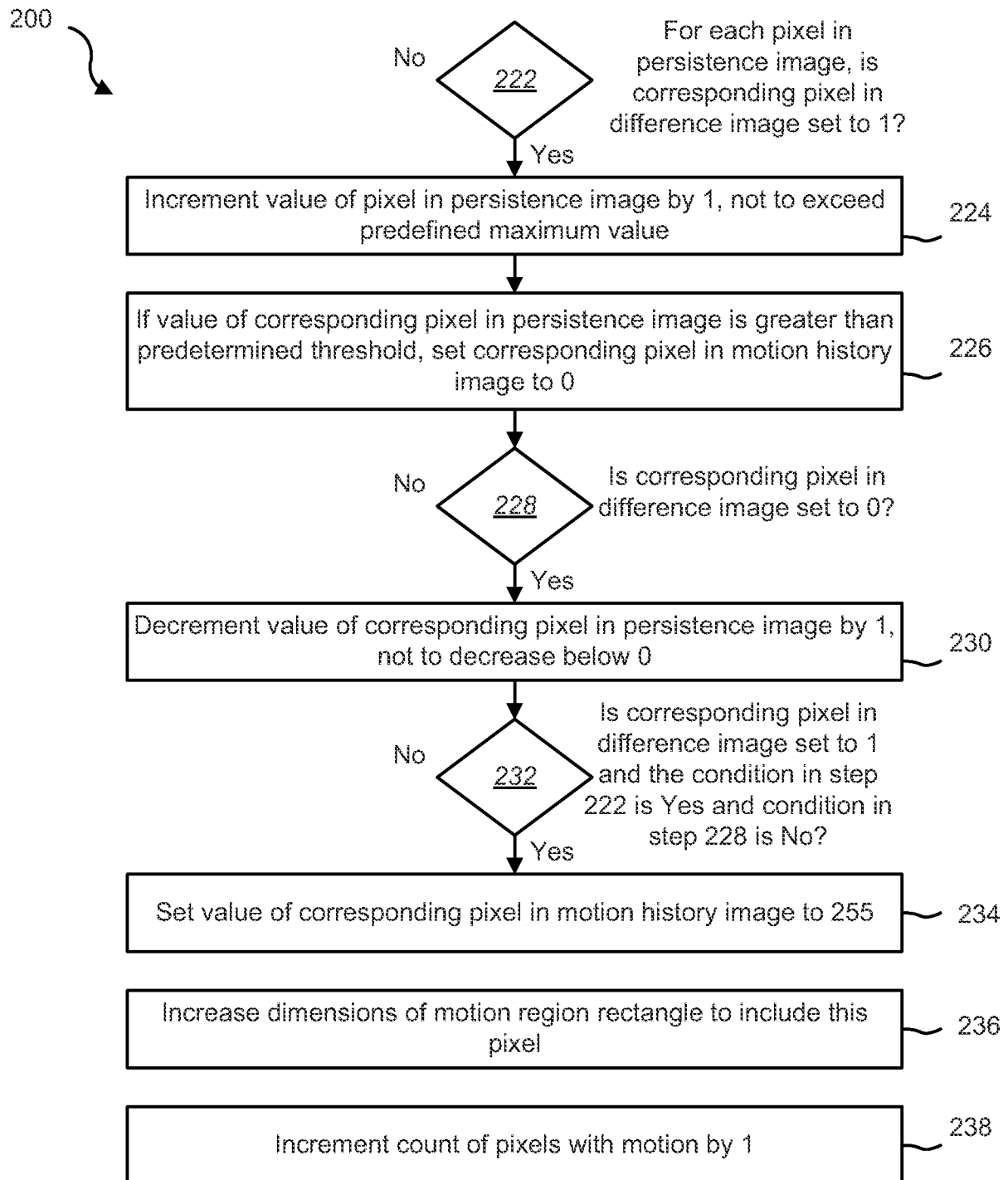
10/19

**FIG. 10**

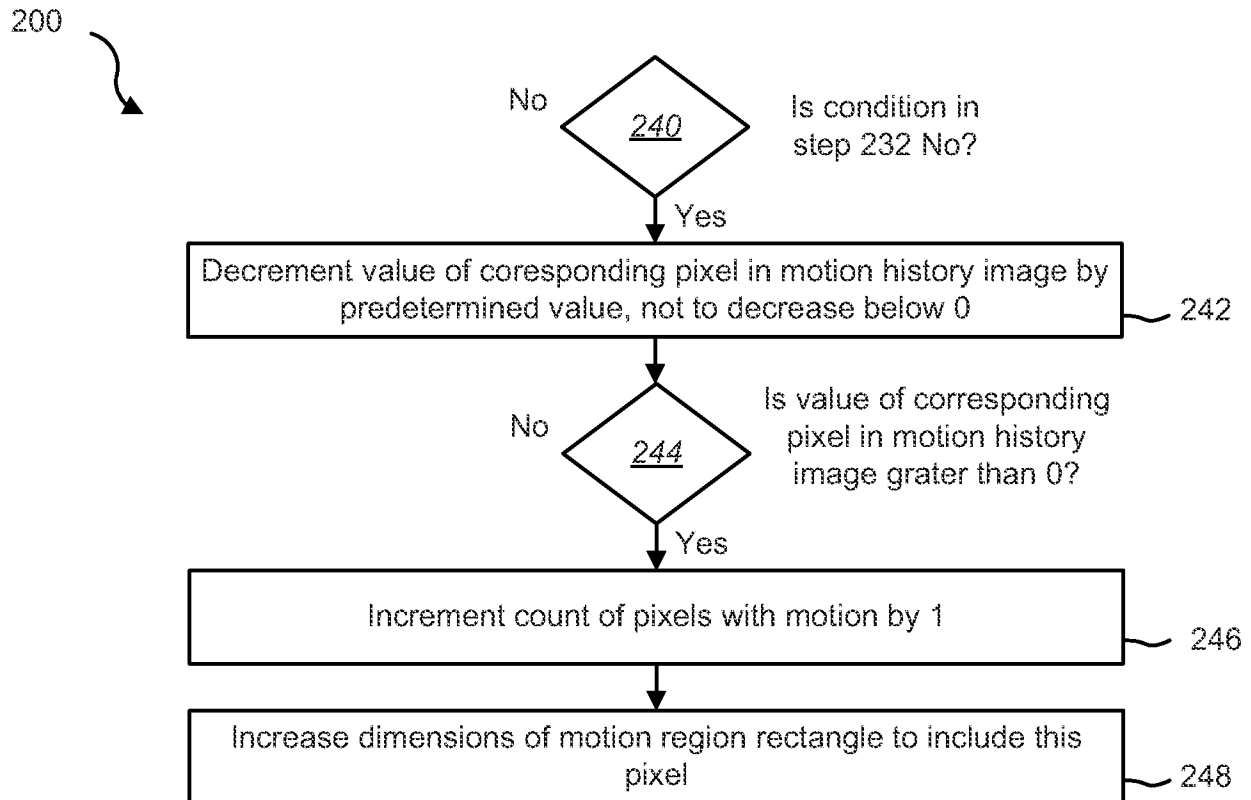
11/19

**FIG. 11**

12/19

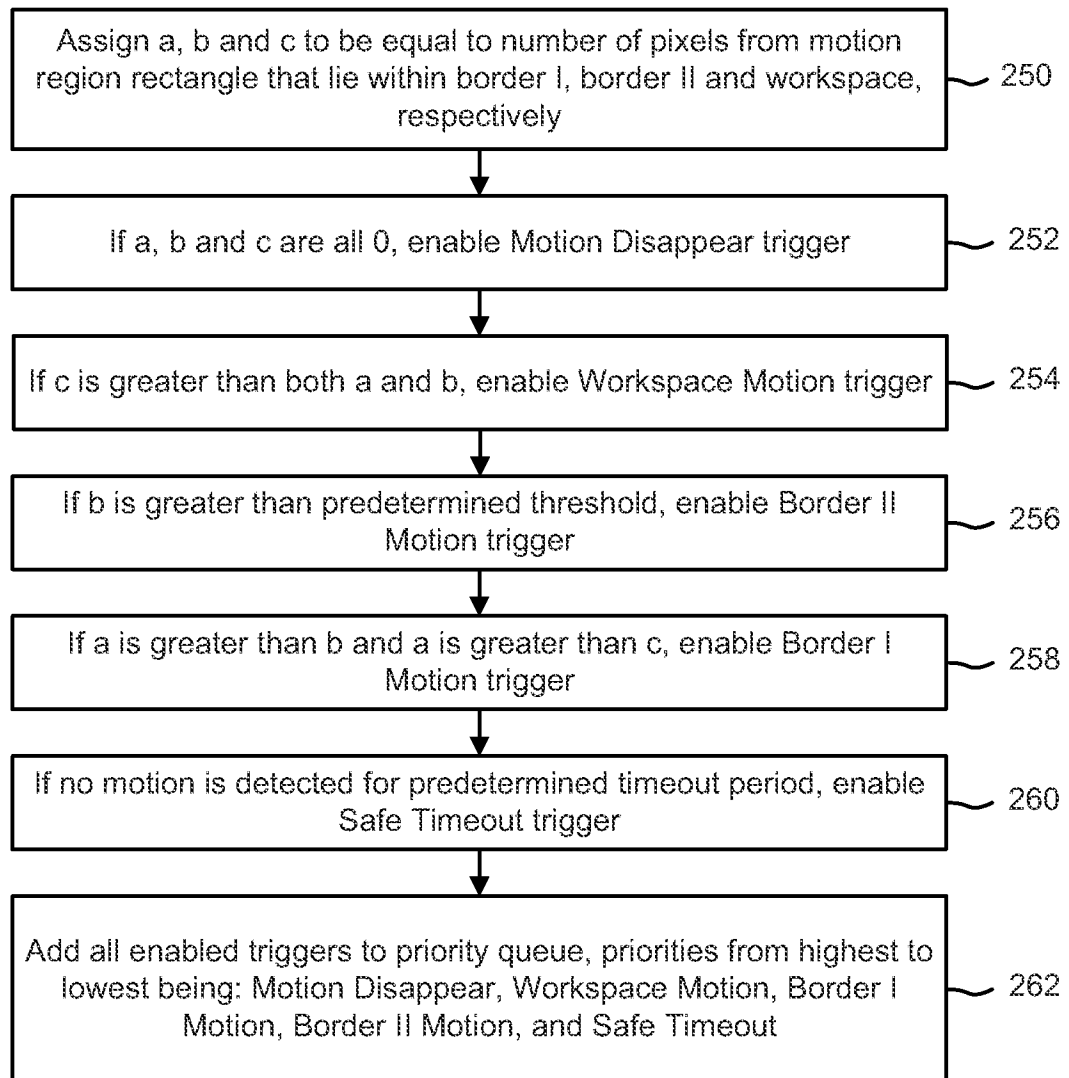
**FIG. 12**

13/19

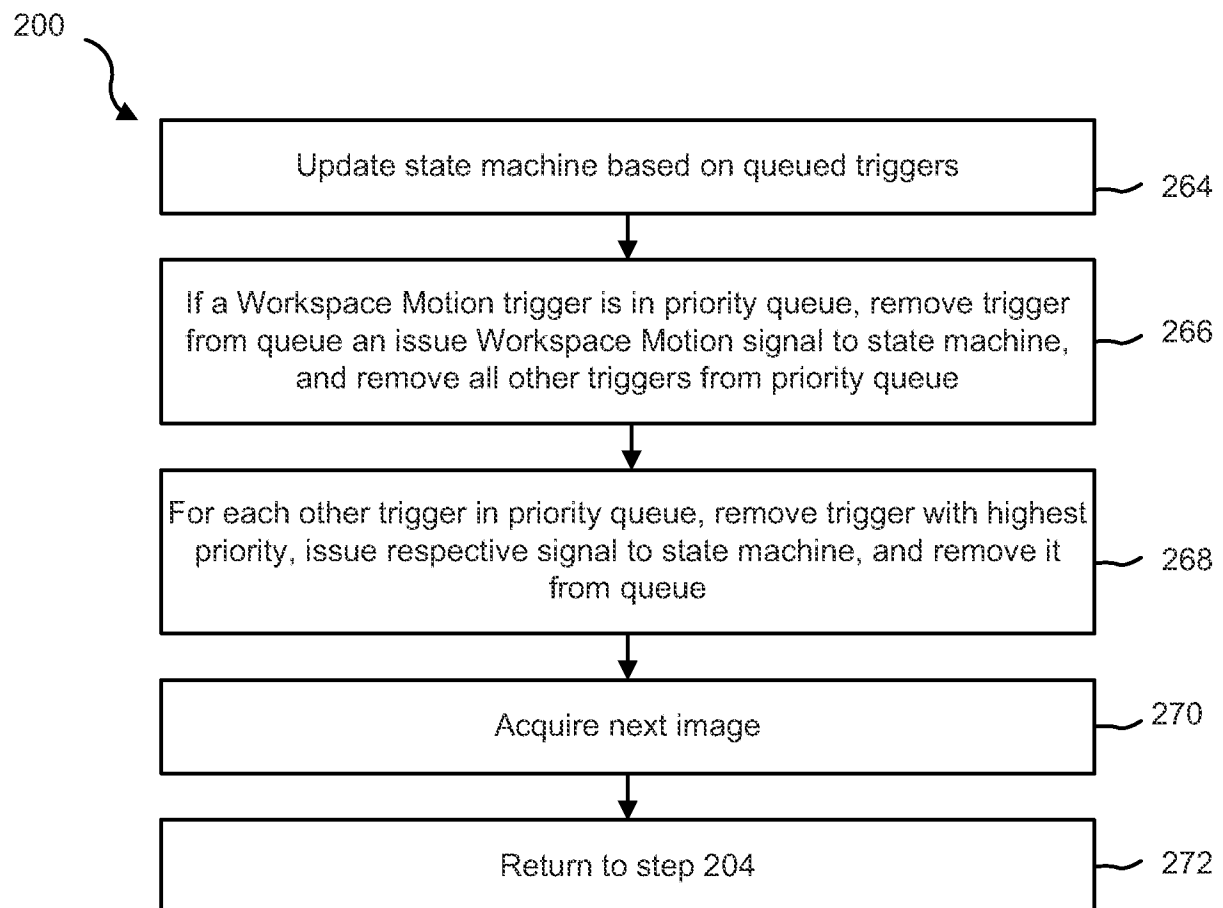
**FIG. 13**

14/19

200

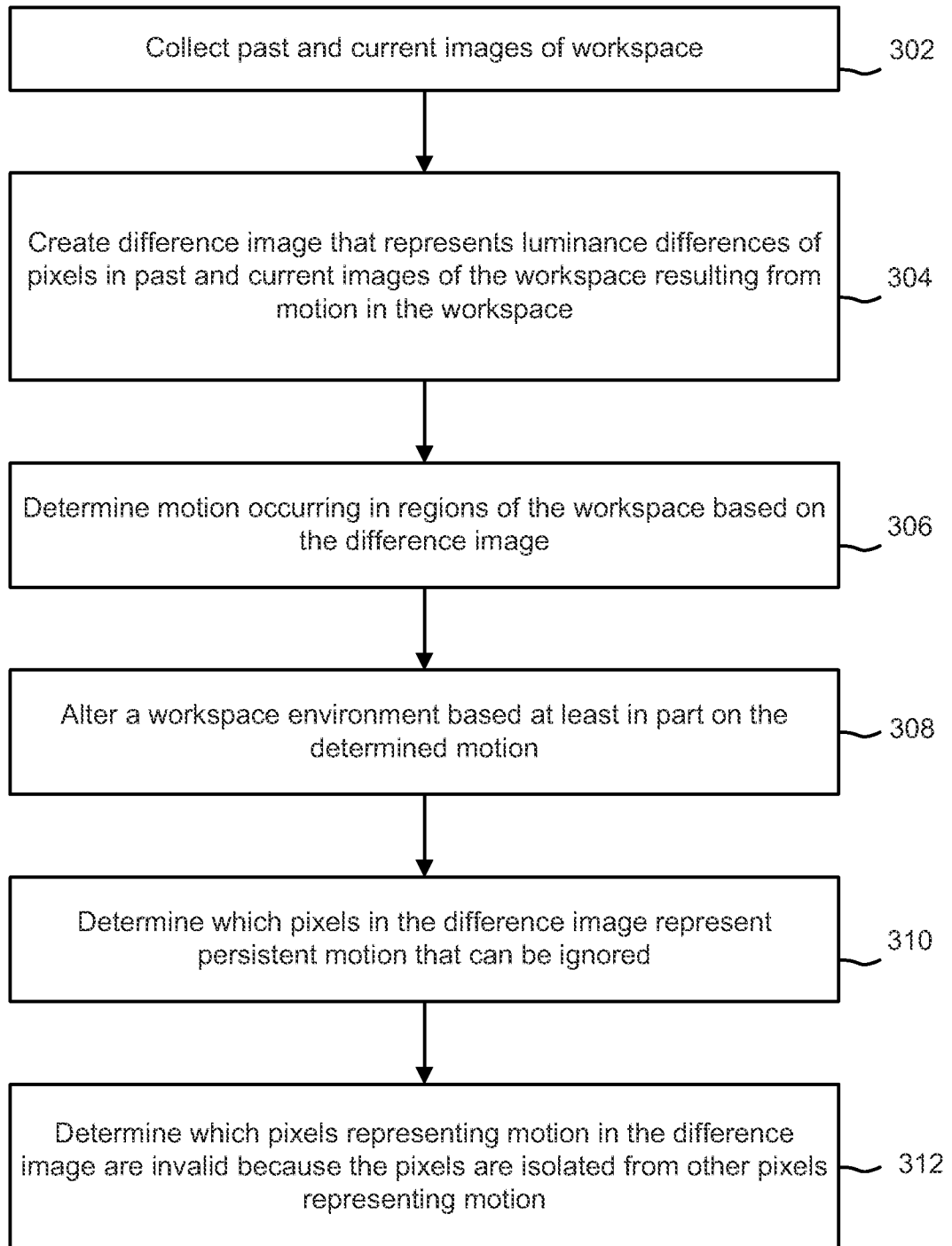
**FIG. 14**

15/19

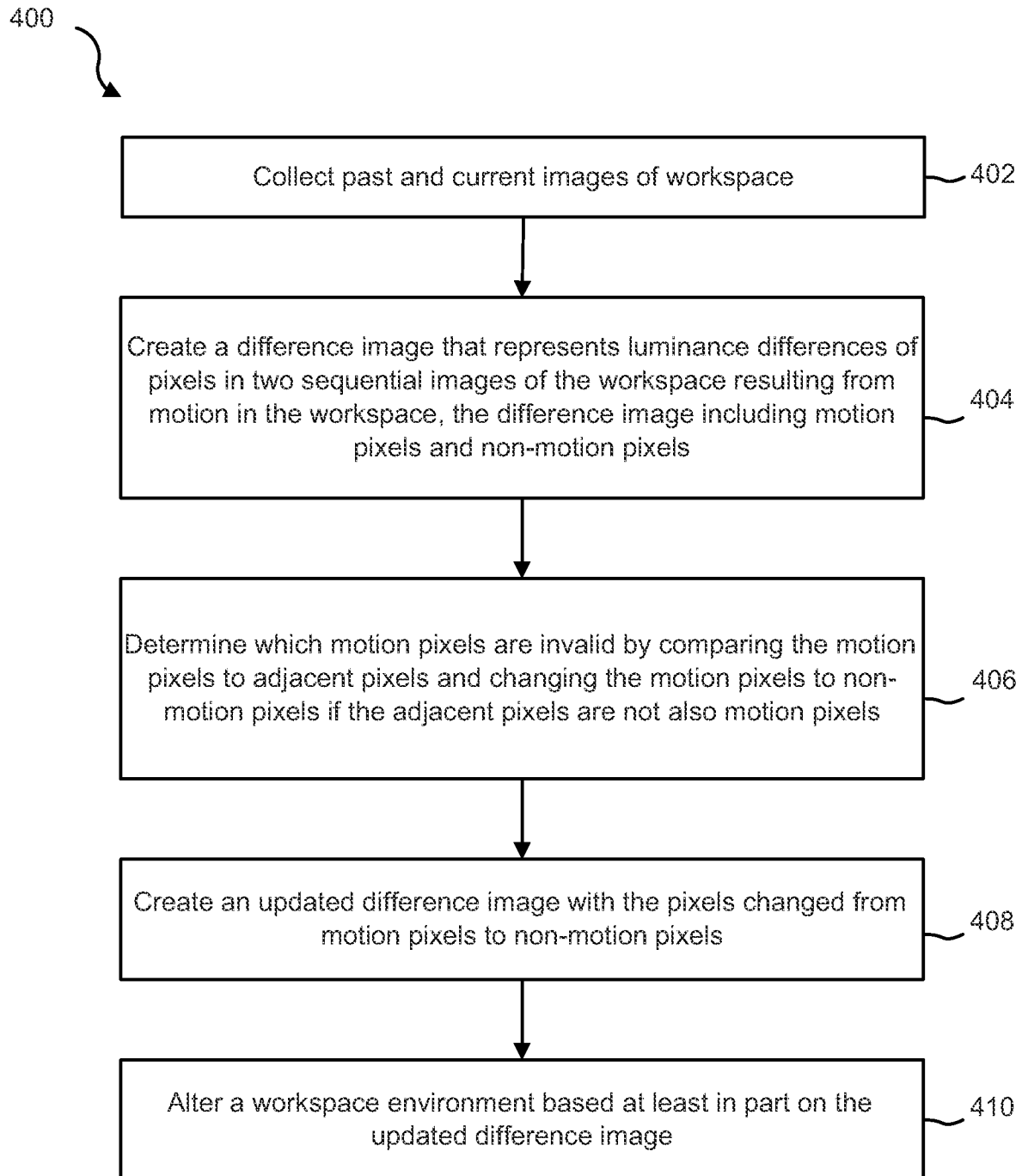
**FIG. 15**

16/19

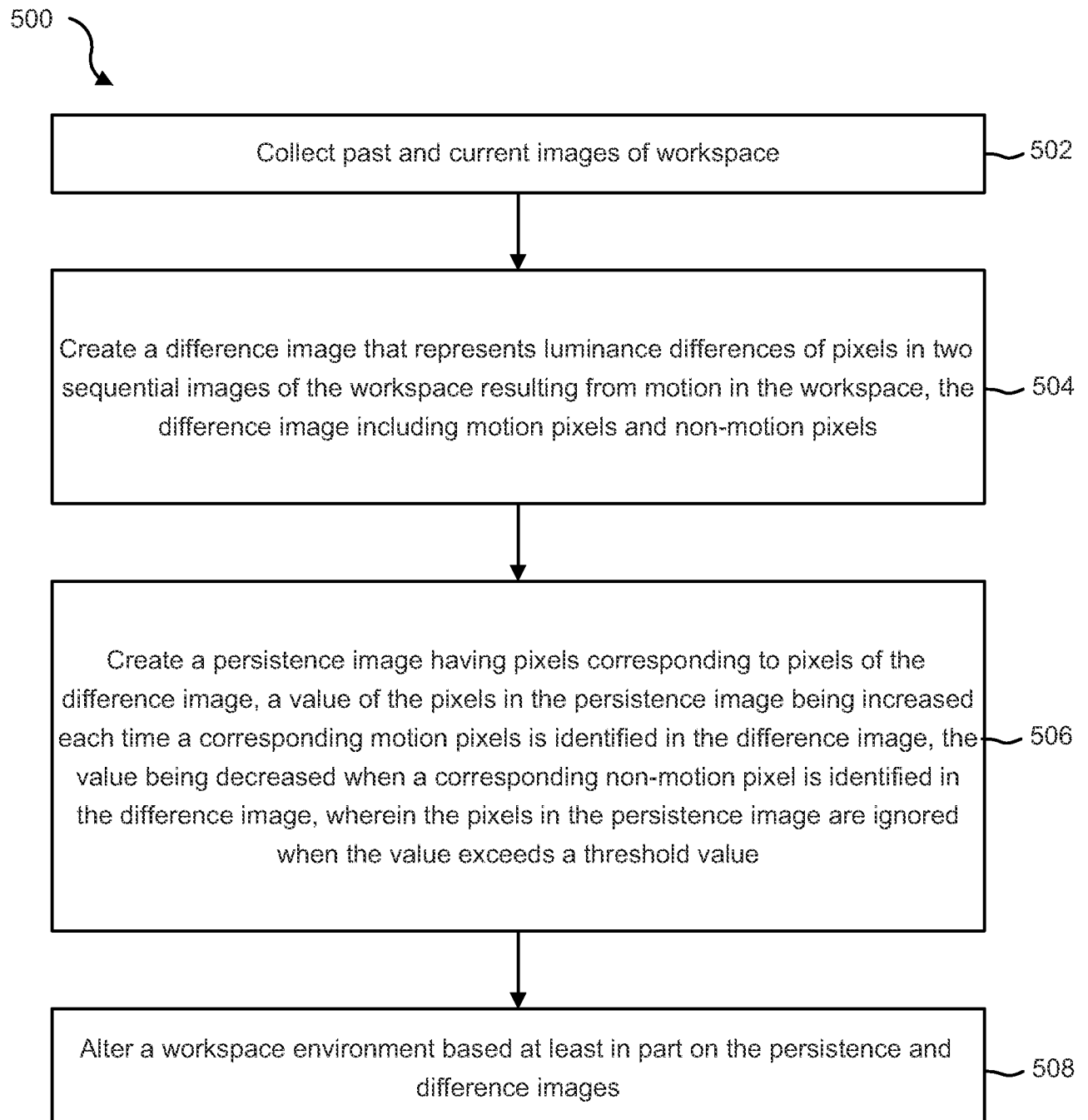
300

**FIG. 16**

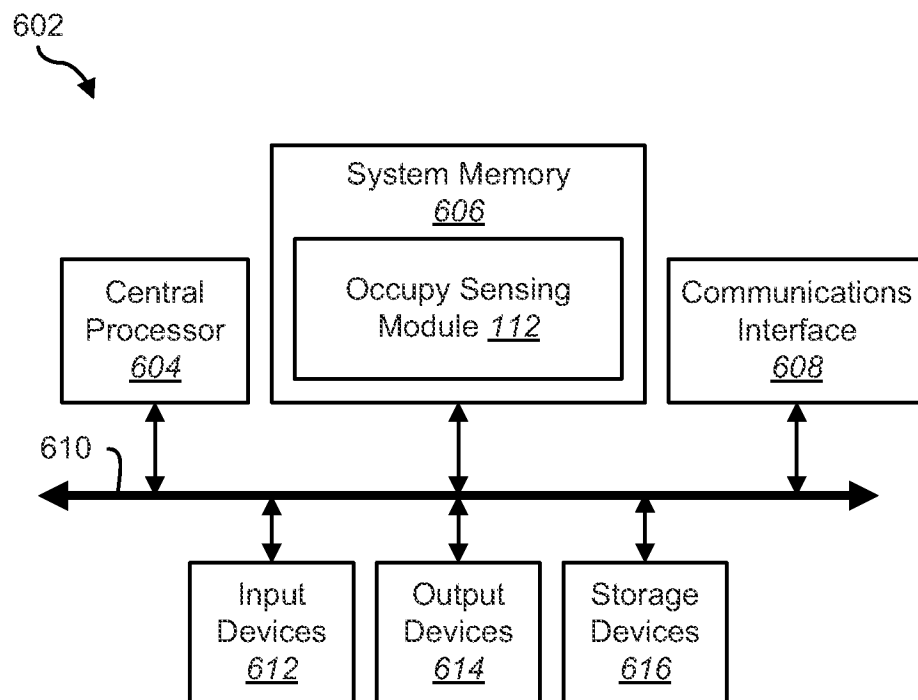
17/19

**FIG. 17**

18/19

**FIG. 18**

19/19

**FIG. 19**