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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 December 2014 (31.12.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/209768 A1

(51) International Patent Classification:
H04N 13/02 (2006.01) ***G06T 15/00*** (2011.01)
G06T 1/60 (2006.01)

(21) International Application Number:
PCT/US2014/043302

(22) International Filing Date:
20 June 2014 (20.06.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/925,459 24 June 2013 (24.06.2013) US

(71) Applicant: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventor: **PFAFFE, Alexander**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

WO 2014/209768 A1

(54) Title: INTERLEAVED TILED RENDERING OF STEREOSCOPIC SCENES

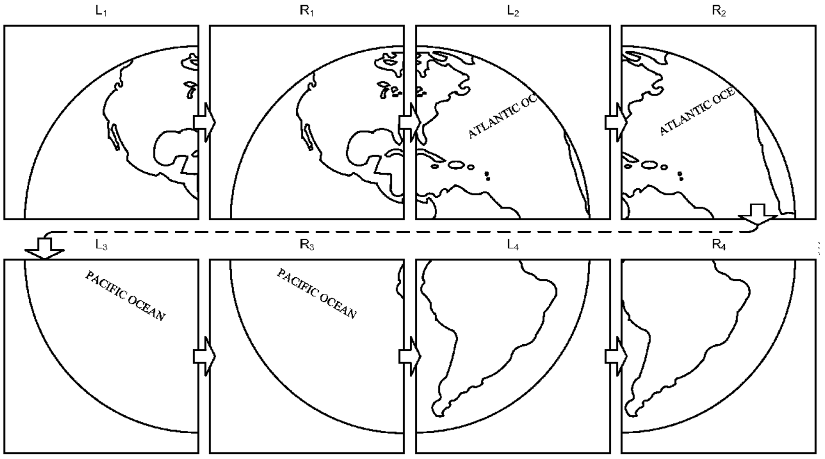

FIG. 6A

(57) Abstract: Embodiments are disclosed that relate to rendering tiles of stereoscopic images in an interleaved manner. For example, one disclosed embodiment provides a method comprising rendering a first tile of an image, and after rendering the first tile of the first image, rendering a first tile of a second image. After rendering the first tile of the second image, a second tile of the first image is rendered, and after rendering the second tile of the first image, a second tile of the second image is rendered. The method further comprises sending the first image to a first eye display, and sending the second image to a second eye display.

INTERLEAVED TILED RENDERING OF STEREOSCOPIC SCENES

BACKGROUND

[0001] In stereoscopic rendering, images of a scene are separately rendered for a user's left and right eyes, wherein perspectives of the left eye image and a perspective of the right eye image are offset similarly to left eye and right eye views of a real world scene. The offset between the left eye image and the right eye image allows the rendered scene to appear as a single three-dimensional scene to a viewer.

SUMMARY

[0002] Embodiments are disclosed that relate to rendering stereoscopic scenes using a tiled renderer. For example, one disclosed embodiment provides a method comprising rendering a first tile of a first image, and after rendering the first tile of the first image, rendering a first tile of a second image. After rendering the first tile of the second image, a second tile of the first image is rendered. After rendering the second tile of the first image, a second tile of the second image is rendered. The method further comprises sending the first image to a first eye display and the second image to a second eye display.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIG. 1 schematically shows an example of a stereoscopically rendered scene being viewed with a head-mounted display device.

[0005] FIG. 2 schematically shows tiles of left and right images of the stereoscopic scene of FIG. 1.

[0006] FIG. 3 shows a block diagram of an embodiment of a memory hierarchy in accordance with the present disclosure.

[0007] FIG. 4 shows a flow diagram depicting an embodiment of a method for rendering tiles of images of a stereoscopic scene in an interleaved manner.

[0008] FIG. 5A schematically shows an order in which tiles of images of a stereoscopic scene are rendered in a non-interleaved manner.

[0009] FIG. 5B shows a graph of errors between successive tile pairs according to the rendering order of FIG. 5A.

[0010] FIG. 6A schematically shows an order in which tiles of images of a stereoscopic scene are rendered in an interleaved manner.

[0011] FIG. 6B shows a graph of errors between successive tile pairs according to the rendering order of FIG. 6A.

[0012] FIG. 7 shows a block diagram of an embodiment of a computing device in accordance with the present disclosure.

DETAILED DESCRIPTION

[0013] In some approaches to rendering three-dimensional graphics, tiled rendering is used to overcome potential issues associated with the hardware used to perform such rendering, such as limited memory bandwidth. Tiled rendering subdivides an image to be rendered into subimages, successively rendering the subimages until the overall image has been rendered for display.

[0014] In stereoscopic rendering, a left and a right image of a scene are separately rendered from different perspectives. When viewed concurrently (or successively at sufficiently high frame rates), the left and right images appear to reproduce the scene in a three-dimensional manner. As two images are rendered, stereoscopic rendering substantially increases (e.g., doubles) the resources utilized to render the three-dimensional scene, including memory bandwidth, time, and power consumption.

[0015] Accordingly, embodiments are disclosed herein that relate to decreasing the resources used to render tiles of stereoscopic images. Briefly, the disclosed embodiments relate to rendering a first tile of a second image after rendering a first tile of a first image and prior to rendering a second tile of the first image. As corresponding tiles of left eye and right eye images may have many similar features, interleaved rendering in this manner may reduce memory access penalties by using at least a portion of data associated with the first tile of the first image to render the first tile of the second image.

[0016] FIG. 1 schematically shows an example of a stereoscopically rendered scene 100 including a stereoscopic object 102 being viewed by a user 104. Stereoscopic scene 100 and stereoscopic object 102 are rendered and displayed in this example by a head-mounted display (HMD) 106 worn by user 104. Here, two images of stereoscopic object 102 are respectively rendered for the left and right eyes of user 104. The images may be respectively rendered from a first perspective and a second perspective that are suitably offset to create a three-dimensional impression.

[0017] In the depicted embodiment, the stereoscopic image is viewed by HMD 106. HMD 106 may represent a virtual reality device having a display that substantially occupies the field of view of user 104, such that user 104 perceives content displayed by such an HMD, and not elements of the surrounding physical environment. In another example, HMD 106 may represent an augmented reality device comprising a see-through display with which images may be displayed over a background physical environment.

[0018] It will be appreciated that HMD 106 is provided merely as an illustrative example and is not intended to be limiting. In other embodiments, stereoscopic scene 100 and stereoscopic object 102 may be presented via a display device not mounted to the head of user 104. For example, a display device may present an image of stereoscopic object 102 which is then partitioned into separate left and right images by polarized lenses in a frame worn by user 104. Alternatively, the display device may alternately display left and right images of stereoscopic object 102 at relatively high speeds (e.g., 120 frames per second). The left and right images may be selectively blocked and transmitted to user 104 via shutter glasses synced to the frame rate of the display output such that only one of the left and right images is perceived at a given instant.

[0019] FIG. 2 shows examples of a left image 202 and a right image 204 of a stereoscopic pair of images. Left and right images 202 and 204 show stereoscopic scene 100 and stereoscopic object 102 from the perspective of the left and right eyes of user 104, respectively. In this example, the first and second perspectives are angularly offset from each other by an offset angle such that a greater leftward portion of stereoscopic object 102 is visible in left image 202, while a greater rightward portion of object 102 is visible in right image 204.

[0020] FIG. 2 also schematically illustrates the tiled rendering of left and right images 202 and 204. As mentioned above, a tiled renderer may help to mitigate hardware constraints that may exist in some devices. For example, a buffer (e.g., frame buffer) to which output from a renderer is written may be too small to store the entirety of the rendered output for a given image of a scene. A tiled renderer thus may be used to subdivide an image of a scene to be rendered into tiles such that the rendered output of a single tile occupies the buffer at any given time. Once written to the buffer, the rendered output for the tile may be sent to a display device before rendering another tile. Alternatively, the rendered output of a given tile may be written to another location in memory (e.g., another buffer) before another tile is rendered. In some implementations, use of a tiled renderer may facilitate rendering parallelism, as each tile may be rendered independently.

[0021] In the depicted example, the tiled renderer has subdivided left and right images 202 and 204 into four equal, rectangular tiles. It will be appreciated, however, that left and right images 202 and 204 may be subdivided into virtually any number of tiles of any suitable shape.

[0022] Left image 202 comprises four tiles successively designated in a clockwise direction: L_1, L_2, L_3 and L_4. Likewise, right image comprises four tiles successively designated in the clockwise direction: R_1, R_2, R_3, and R_4. As shown, each set of four tiles for a corresponding image (e.g., L_1, L_2, L_3 and L_4 for left image 202) includes substantially different elements of stereoscopic scene 100 and stereoscopic object 102. In contrast, spatially corresponding tile pairs between left and right images 202 and 204 (e.g., L_1 and R_1, L_2 and R_2, L_3 and R_3, and L_4 and R_4) include substantially similar elements of stereoscopic scene 100 and stereoscopic object 102, as they correspond to substantially similar regions of the scene and object but have an angular offset, as described above. Such tile pairs may be said to be substantially spatially coherent. The spatial coherence of such tile pairs may be leveraged to reduce the time, power, and memory access associated with rendering left and right images 202 and 204 as described in further detail below with reference to FIGS. 4, 6A, and 6B.

[0023] FIG. 3 shows an example memory hierarchy 300 that may be utilized in a tile-based rendering pipeline for rendering left and right images 202 and 204. Hierarchy 300 includes main memory 302. Main memory 302 may have the highest capacity, but also the highest latency, wherein “latency” refers to the time at which data is available following a request for that data in memory. Prior to or during execution of the rendering pipeline with which left and right images 202 and 204 are rendered, data used for rendering stereoscopic scene 100 and stereoscopic object 102 may be written to main memory 302. Such scene data may include, for example, rendering engine and other application code, primitive data, textures, etc.

[0024] Memory hierarchy 300 further includes a command buffer 304 operatively coupled to main memory 302 via a bus, represented in FIG. 3 by a dashed line. In this example, command buffer 304 occupies a smaller, separate region of memory and may have a reduced latency compared to that of main memory 302. Requests for data in command buffer 304 may thus be satisfied in a shorter time. Data for one (or in some embodiments, both) of left and right images 202 and 204 may be written to command buffer 304 from main memory 302 such that the data may be accessed by the rendering pipeline in an expedited manner. This data may include the command programs, associated parameters, and any other resources required to render the image, including but not limited to shaders, constants, textures, a vertex buffer, index buffer, and a view transformation matrix or other data structure encoding information regarding a perspective from which the image (e.g., left image 202) is to be rendered.

[0025] Memory hierarchy 300 also includes a tile buffer 306 operatively coupled to command buffer 304 via a bus represented by a dashed line. Tile buffer 306 may occupy a smaller, separate region of memory and may have a reduced latency compared to that of command buffer 304. Data for a particular tile (e.g., L_1) may be written to tile buffer 306 from command buffer 304 such that data for the particular tile may be accessed by the rendering pipeline and the tiled renderer in a further expedited manner. Tile buffer 306 may be configured to store an entirety of tile data for a given tile and a given tile size.

[0026] In some embodiments, command buffer 304 and tile buffer 306 occupy regions of a first cache and a second cache respectively allocated to the buffers. The first cache may have a first latency, while the second cache may have a second latency which may be less than the first latency. In this way, memory fetches for tile data may be optimized and latency penalties resulting from tile data fetches reduced.

[0027] It will be appreciated that main memory 302, command buffer 304, and tile buffer 306 may each correspond to a discrete, physical memory module which may be operatively coupled to a logic device. Alternatively, one or more of main memory 302, command buffer 304, and tile buffer 306 may correspond to a single physical memory module, and may be further embedded with a logic device in a system-on-a-chip (SoC) configuration. Moreover, the busses which facilitate reads and writes among main memory 302, command buffer 304, and tile buffer 306 are exemplary in nature. In other embodiments, for example, tile buffer 306 may be operatively and directly coupled to main memory 302.

[0028] FIG. 4 shows a flow diagram depicting an embodiment of a method 400 for rendering tiles of images of a stereoscopic scene in an interleaved manner. Method 400 is described with reference to stereoscopic scene 100 left and right images 202 and 204 and their constituent tiles, and memory hierarchy 300. However, it will be understood that the method may be used in any other tiled rendering scenario and hardware environment in which a common scene is rendered from two or more perspectives. Examples of suitable hardware are described in more detail below with reference to FIG. 7.

[0029] At 402, method 400 comprises writing scene data for stereoscopic scene 100 to command buffer 304 from main memory 302. As described above, the scene data may comprise a plurality of elements for rendering stereoscopic scene 100 and stereoscopic object 102, such as primitives which model the substantially spherical shape of the object, textures which affect the surface appearance of the object, etc. It will be appreciated that prior to 402, such scene data, along with other data such as rendering pipeline and other application code, may be written to main memory 302 such that command and tile buffers 304 and 306 may read the scene data from main memory.

[0030] At 404, method 400 comprises extracting first tile data for a first image from the scene data written to command buffer. For example, tile data associated with tile L_1 of left image 202 may be extracted from the scene data. The tile data may be a subset of the scene data, comprising primitives, textures, etc. corresponding to the first tile but not other tiles of the left image. Extraction of the first tile data may include actions such as a clipping, scissor, or occlusion culling operation to determine the scene data specific to the first tile. Method 400 further comprises, at 406, writing the first tile data for the first image to tile buffer 306.

[0031] At 408, the first tile (e.g., L_1) of the first image (e.g., left image 202) is rendered. As described above, rendering may include transformation, texturing, shading, etc. which collectively translate first tile data into data which may be sent to a display device (e.g., HMD 106) to produce an observable image (e.g., stereoscopic scene 100 observed by user 104).

[0032] Next, at 410, a first tile (e.g., R_1) of a second image (e.g., right image 204) is rendered based on the tile data previously written to and currently occupying tile buffer 306 for the first image tile. Here, the potentially substantial spatial coherence between the spatially corresponding tile pair L_1-R_1 is utilized, as a significant portion of L_1 already written to tile buffer 306 may be reused to render R_1. In this way, the time, processing resources, power, etc., which might be otherwise doubled in rendering two dissimilar image tiles of a stereoscopic scene, may be reduced. More particularly, rendering a tile (e.g., R_1) of a second image (e.g., right image 204) after rendering a spatially corresponding tile (e.g., L_1) of a first image (e.g., left image 202) may result in a reduced number of memory fetches to command buffer 304, compared to rendering all tiles (e.g., L_1-L_4) of the first image before rendering the first tile (e.g., R_1) of the second image.

[0033] It will be appreciated that, prior to performing process 410, the view transformation matrix described above, which may reside in command buffer 304, may be utilized to redetermine the perspective from which the second image (e.g., right image 204) is rendered.

[0034] In some instances, some data used for rendering of the first tile of the second image may not be in the tile buffer (e.g. due to the slightly different perspectives of stereoscopic images). Thus, after rendering at least a portion of the first tile of the second image using the tile data in the tile buffer, and prior to rendering the second tile of the first image, a remaining portion of the first tile of the second image may be rendered based on tile data in the command buffer if a tile buffer miss occurs during rendering of the first tile of the second image. This is illustrated at 412, where tile data (e.g., data for R_1) for the second image (e.g., right image 204) is obtained from command buffer 304 if there is a miss for the tile data in tile buffer 306. In embodiments in which tile buffer 306 occupies a cache, the tile buffer miss corresponds to a cache miss. In some scenarios, access to command buffer 304 may be omitted, as the tile data already written to tile buffer 306 at 406 is sufficient to fully render this tile.

[0035] At 414, method 400 comprises determining whether there are additional tiles for the first and second images which have yet to be rendered. If there are no additional tiles for the first and second images which have yet to rendered, then method 400 proceeds to 418, where the first image is sent to the first eye display and the second image is sent to the second eye display.

[0036] On the other hand, if there are additional tiles for the first and second images which have yet to rendered, then method 400 proceeds to 416 where tile data for the next tile (e.g., L_2) for the first image (e.g., left image 202) is extracted from scene data in command buffer 304 as at 404. Following tile data extraction for the next tile for the first image, the next tile is rendered as at 408. Method 400 thus proceeds iteratively until all tiles of the first and second images have been rendered, at which point the first and second images are sent respectively to the first eye display and the second eye display. It will be appreciated that the first and second images sent to respective eye displays at 418 and 420 may be performed concurrently or successively as described above. Further, the first and second eye displays may be separate display devices or form a contiguous display device, and may be part of a wearable display device such as HMD 106 or a non-wearable display device such as a computer display (e.g. monitor, tablet screen, smart phone screen, laptop screen, etc.).

[0037] The potential savings of computing resources that may be achieved by following method 400 is demonstrated via FIGS. 5A-5B and FIGS. 6A-6B, wherein FIGS. 5A and 5B show non-interleaved tiled rendering of stereoscopic images and FIGS. 6A-6B show an example of tiled rendering of stereoscopic images according to method 400.

[0038] First regarding FIGS. 5A-5B, the tile set for left image 202 is successively rendered in the order L_1, L_2, L_3, L_4. After the tile set for left image 202 has been rendered, the entire tile set for right image 204 is successively rendered in the order R_1, R_2, R_3, R_4. In this approach, the spatial coherency between spatially corresponding tile pairs of two images is not leveraged. As such, a substantially entire set of data is written to the tile buffer for each rendered tile. Consequently, stereoscopic rendering of two images of a scene (e.g., scene 100) in this approach may utilize roughly double the computing resources compared to rendering of a single image of the same scene.

[0039] FIG. 5B shows a graph 550 of image error computed between each successive tile pair of left and right images 202 and 204 according to the order in which the tiles are rendered based on the approach represented in FIG. 5A. As illustrated, the error between each successive tile pair may fluctuate about a relatively high error value, indicating significant differences in the image content between successive tile pairs. It will be appreciated that error graph 550 is provided as an illustrative example, and that similar error graphs produced using the rendering approach of FIG. 5A may display greater or lesser errors between successive tiles depending on the visual content of the tiles being rendered.

[0040] Error graph 550 may also represent a relative amount of data copied from command buffer 304 to tile buffer 306 when rendering the second tile of each adjacent pair of tiles – for example, the error value corresponding to the L_4, R_1 pair may represent the amount of data copied to the tile buffer when rendering tile R_1 after rendering tile L_4. Although in some instances a portion of the second tile of each tile pair may be rendered based on data residing in tile buffer 306 previously written in order to render the first tile (e.g., corresponding to a cache hit), in this example a majority of the tile data required to render the second tile is copied from command buffer 304 to tile buffer 306 (e.g., corresponding to a cache miss).

[0041] Next, FIG. 6A shows the order in which the tiles of left and right images 202 and 204 of FIG. 2 are rendered according to method 400 of FIG. 4. Here, tiles are rendered in an interleaved manner based on spatial coherence in the following manner: L_1, R_1, L_2, R_2, L_3, R_3, L_4, and R_4. By rendering the tiles of left and right images 202 and 204 in this order, tile data already written to tile buffer 306 (FIG. 3) in order to render the first tile of a spatially corresponding tile pair is leveraged when rendering the second tile of the tile pair. Accordingly, the computational cost (e.g., time, power, etc.) incurred during rendering images of a stereoscopic scene may be significantly reduced, and in some instances potentially by a factor of close to two.

[0042] FIG. 6B shows a graph 650 illustrating image error computed between adjacent tiles according to the order in which the tiles are rendered based on the approach of FIGS. 4 and 6A. Due to interleaved rendering, the error alternates between a greater relative error and a lesser relative error (relative to each other), starting at a lesser error due to the spatial correspondence between tiles L_1 and R_1.

[0043] Graph 650 may also represent the amount of data copied from command buffer 304 to tile buffer 306 (FIG. 3) when rendering the second tile of each pair. For example, a relatively low amount of data is copied to tile buffer 306 when rendering tile R_1 following rendering tile L_1, as a substantial portion of tile data previously copied to the tile buffer for rendering tile L_1 is reused to render R_1. Conversely, a relatively low amount of data previously written to and residing in tile buffer 306 may be leveraged when rendering tile pairs which do not spatially correspond – for example when rendering tile L_2 after rendering tile R_1.

[0044] Thus, the disclosed embodiments may allow for the efficient use of computing resources when performing tiled rendering of stereoscopic images. In some embodiments, the methods and processes described herein may be tied to a computing system of one or more computing devices. In particular, such methods and processes may be implemented as a computer-application program or service, an application-programming interface (API), a library, and/or other computer-program product.

[0045] FIG. 7 schematically shows a non-limiting embodiment of a computing system 700 that can enact one or more of the methods and processes described above. Computing system 700 is shown in simplified form. Computing system 700 may take the form of one or more personal computers, server computers, tablet computers, home-entertainment computers, network computing devices, gaming devices, mobile computing devices, mobile communication devices (e.g., smart phone), and/or other computing devices.

[0046] Computing system 700 includes a logic subsystem 702 and a storage subsystem 704. Computing system 700 may optionally include a display subsystem 706, input subsystem 708, communication subsystem 710, and/or other components not shown in FIG. 7.

[0047] Logic subsystem 702 includes one or more physical devices configured to execute instructions. For example, the logic subsystem may be configured to execute instructions that are part of one or more applications, services, programs, routines, libraries, objects, components, data structures, or other logical constructs. Such instructions may be implemented to perform a task, implement a data type, transform the state of one or more components, achieve a technical effect, or otherwise arrive at a desired result.

[0048] The logic subsystem may include one or more processors configured to execute software instructions. Additionally or alternatively, the logic subsystem may include one or more hardware or firmware logic subsystems configured to execute hardware or firmware instructions. Processors of the logic subsystem may be single-core or multi-core, and the instructions executed thereon may be configured for sequential, parallel, and/or distributed processing. Individual components of the logic subsystem optionally may be distributed among two or more separate devices, which may be remotely located and/or configured for coordinated processing. Aspects of the logic subsystem may be virtualized and executed by remotely accessible, networked computing devices configured in a cloud-computing configuration.

[0049] Storage subsystem 704 includes one or more physical devices comprising computer-readable storage media configured to hold instructions executable by the logic subsystem to implement the methods and processes described herein. When such methods and processes are implemented, the state of storage subsystem 704 may be transformed—e.g., to hold different data.

[0050] Storage subsystem 704 may include removable and/or built-in devices. Storage subsystem 704 may include optical memory (e.g., CD, DVD, HD-DVD, Blu-Ray Disc, etc.), semiconductor memory (e.g., RAM, EPROM, EEPROM, etc.) including memory hierarchy 300 of FIG. 3, one or more caches (e.g., level 1 cache, level 2 cache, etc.) and/or magnetic memory (e.g., hard-disk drive, floppy-disk drive, tape drive, MRAM, etc.), among others. Storage subsystem 704 may include volatile, nonvolatile, dynamic, static, read/write, read-only, random-access, sequential-access, location-addressable, file-addressable, and/or content-addressable devices.

[0051] It will be appreciated that storage subsystem 704 includes one or more physical devices and excludes propagating signals per se. However, aspects of the instructions described herein alternatively may be propagated by a communication medium (e.g., an electromagnetic signal, an optical signal, etc.), as opposed to being stored in a computer-readable storage medium.

[0052] Aspects of logic subsystem 702 and storage subsystem 704 may be integrated together into one or more hardware-logic components. Such hardware-logic components may include field-programmable gate arrays (FPGAs), program- and application-specific integrated circuits (PASIC / ASICs), program- and application-specific standard products (PSSP / ASSPs), system-on-a-chip (SOC), and complex programmable logic devices (CPLDs), for example.

[0053] The term "program" may be used to describe an aspect of computing system 700 implemented to perform a particular function. In some cases, a program may be instantiated via logic subsystem 702 executing instructions held by storage subsystem 704. It will be understood that different programs may be instantiated from the same application, service, code block, object, library, routine, API, function, etc. Likewise, the same program may be instantiated by different applications, services, code blocks, objects, routines, APIs, functions, etc. The term “program” may encompass individual or groups of executable files, data files, libraries, drivers, scripts, database records, etc.

[0054] Display subsystem 706 may be used to present a visual representation of data held by storage subsystem 704. As the herein described methods and processes change the data held by the storage machine, and thus transform the state of the storage machine, the state of display subsystem 706 may likewise be transformed to visually represent changes in the underlying data. Display subsystem 706 may include one or more display devices utilizing virtually any type of technology, including but not limited to HMD 106 of FIG. 1. Such display devices may be combined with logic subsystem 702 and/or storage subsystem 704 in a shared enclosure, or such display devices may be peripheral display devices.

[0055] When included, input subsystem 708 may comprise or interface with one or more user-input devices such as a keyboard, mouse, touch screen, or game controller. In some embodiments, the input subsystem may comprise or interface with selected natural user input (NUI) componentry. Such componentry may be integrated or peripheral, and the transduction and/or processing of input actions may be handled on- or off-board. Example NUI componentry may include a microphone for speech and/or voice recognition; an infrared, color, stereoscopic, and/or depth camera for machine vision and/or gesture recognition; a head tracker, eye tracker, accelerometer, and/or gyroscope for motion detection and/or intent recognition; as well as electric-field sensing componentry for assessing brain activity.

[0056] When included, communication subsystem 710 may be configured to communicatively couple computing system 700 with one or more other computing devices. Communication subsystem 710 may include wired and/or wireless communication devices compatible with one or more different communication protocols. As non-limiting examples, the communication subsystem may be configured for communication via a wireless telephone network, or a wired or wireless local- or wide-area network. In some embodiments, the communication subsystem may allow computing system 700 to send and/or receive messages to and/or from other devices via a network such as the Internet.

[0057] It will be understood that the configurations and/or approaches described herein are exemplary in nature, and that these specific embodiments or examples are not to be considered in a limiting sense, because numerous variations are possible. The specific routines or methods described herein may represent one or more of any number of processing strategies. As such, various acts illustrated and/or described may be performed in the sequence illustrated and/or described, in other sequences, in parallel, or omitted. Likewise, the order of the above-described processes may be changed.

[0058] The subject matter of the present disclosure includes all novel and nonobvious combinations and subcombinations of the various processes, systems and configurations, and other features, functions, acts, and/or properties disclosed herein, as well as any and all equivalents thereof.

CLAIMS

1. On a computing device, a method for producing a stereoscopic image using a tiled renderer, the stereoscopic image comprising a first image of a scene from a first perspective and a second image of the scene from a second perspective, the method comprising:

rendering a first tile of the first image;

after rendering the first tile of the first image, rendering a first tile of the second image;

after rendering the first tile of the second image, rendering a second tile of the first image;

after rendering the second tile of the first image, rendering a second tile of the second image; and

sending the first image to a first eye display and sending the second image to the second eye display.

2. The method of claim 1, further comprising:

prior to rendering the first tile of the first image, copying data associated with the first image and the second image to a command buffer in a memory cache.

3. The method of claim 2, wherein rendering the first tile of the second image results in a reduced number of memory fetches to the command buffer compared to rendering all tiles of the first image before rendering the first tile of the second image.

4. The method of claim 2, further comprising:

prior to rendering the first tile of the first image, and after copying the data to the command buffer, copying data associated with the first tile of the first image from the command buffer to a tile buffer.

5. The method of claim 4, further comprising:

wherein the command buffer has a first latency and the tile buffer has a second latency, the second latency less than the first latency.

6. The method of claim 4, wherein a remaining portion of the first tile of the second image is rendered based on data in the command buffer if a tile buffer miss occurs.

7. The method of claim 2, wherein the first tile of the second image is rendered at least partially based on the data associated with the first tile of the first image.

8. The method of claim 1, wherein an error computed between successive pairs of tiles alternates between a greater error and a lesser error.

9. The method of claim 1, wherein the first perspective at least partially overlaps the second perspective.

10. The method of claim 1, wherein the first tile of the first image spatially corresponds to the first tile of the second image.

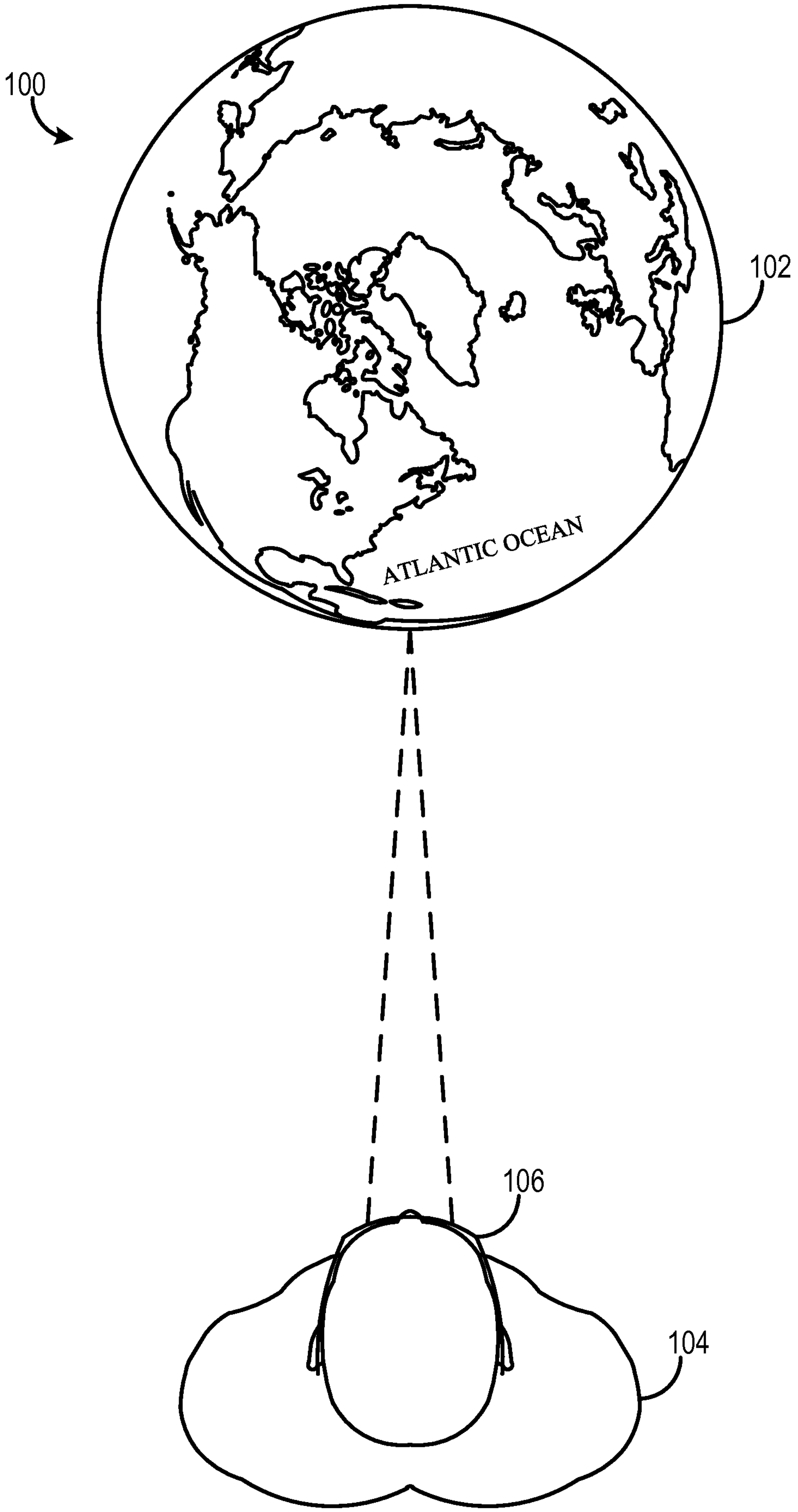

FIG. 1

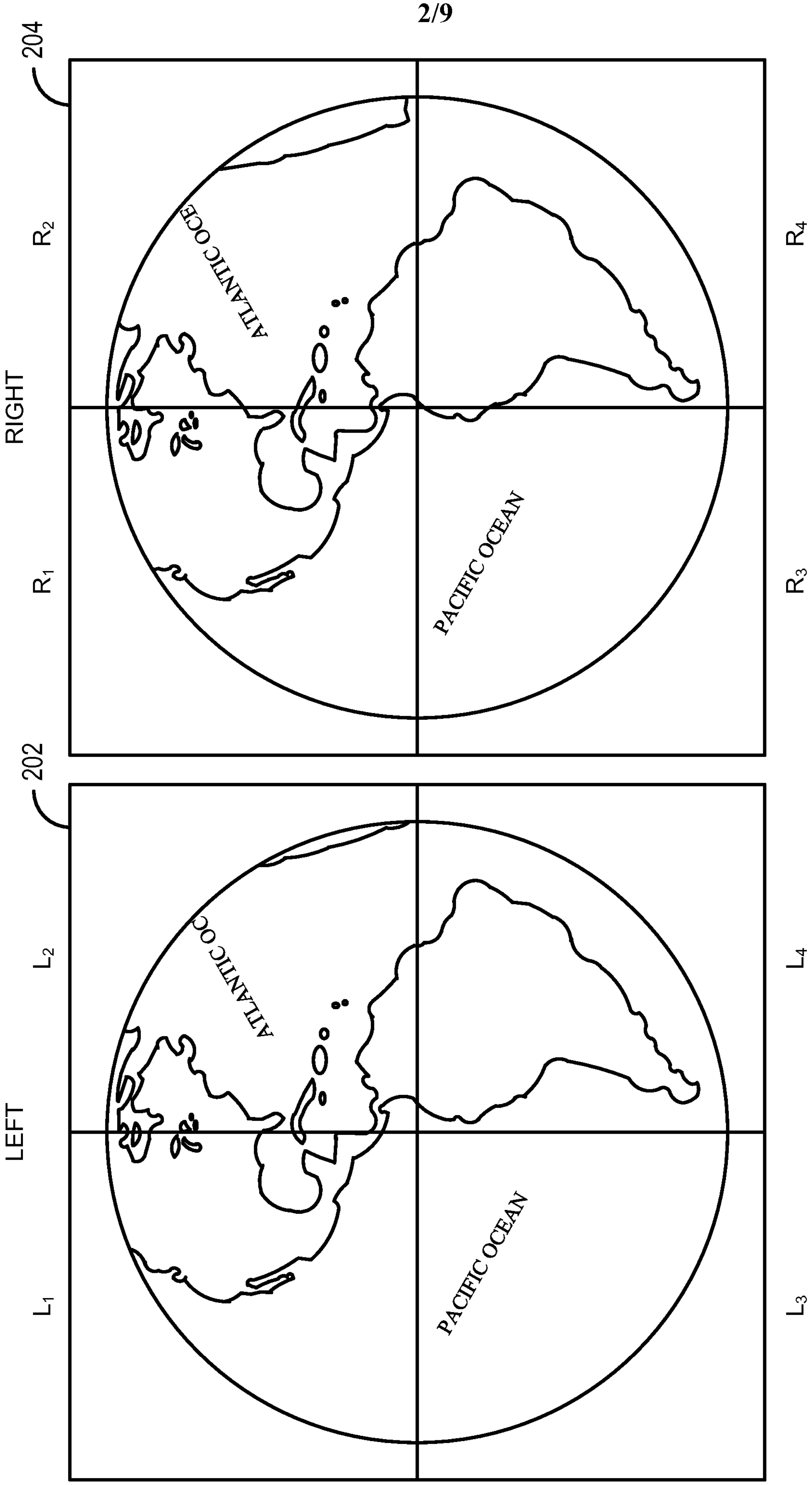
204
RIGHT
R_2
R_1
R_4
R_3
ATLANTIC OCE
PACIFIC OCEAN
202
LEFT
L_2
L_1
L_4
L_3
ATLANTIC OC
PACIFIC OCEAN

FIG. 2

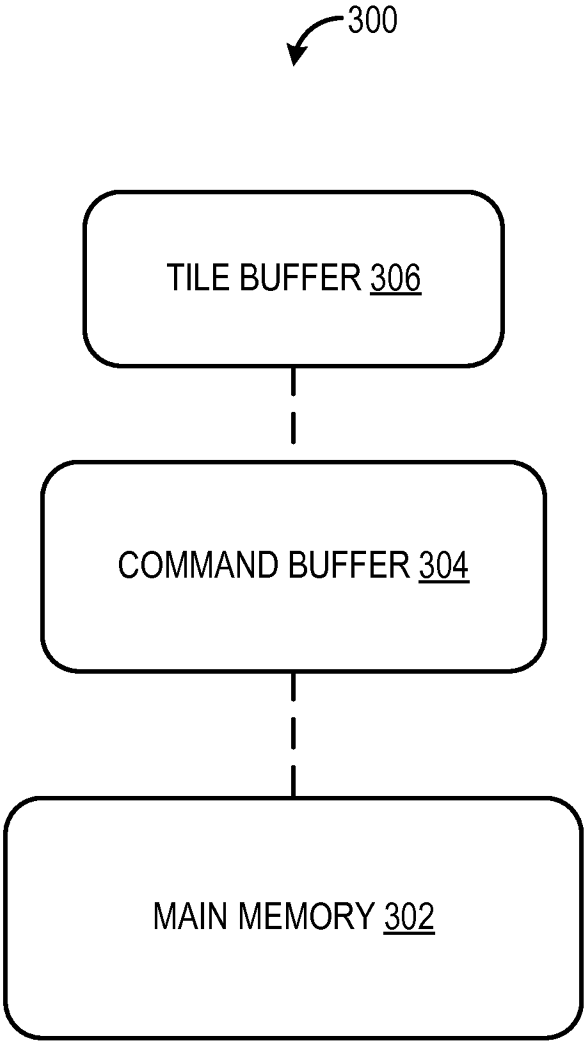

FIG. 3

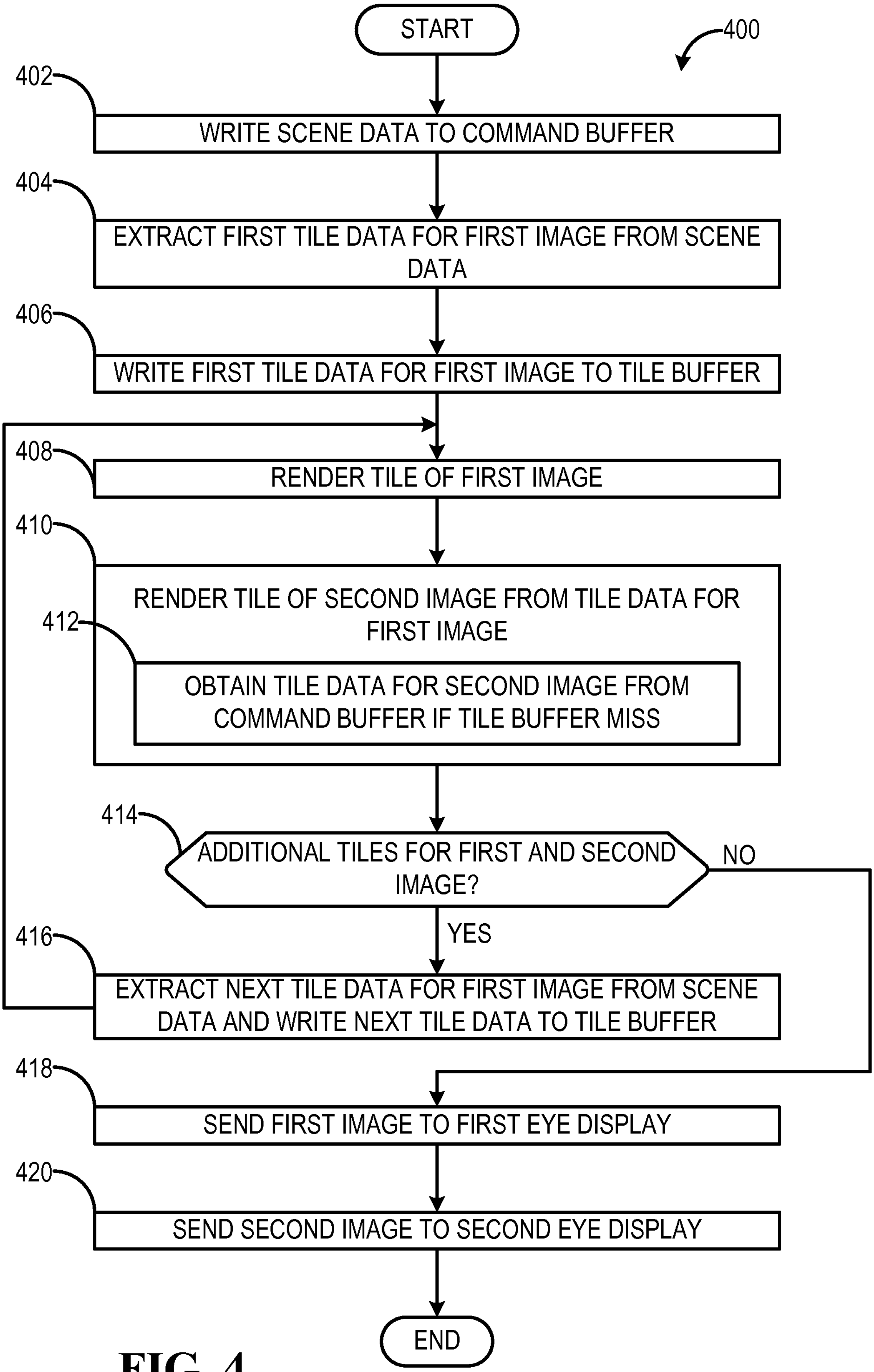
START
400
402
WRITE SCENE DATA TO COMMAND BUFFER
404
EXTRACT FIRST TILE DATA FOR FIRST IMAGE FROM SCENE DATA
406
WRITE FIRST TILE DATA FOR FIRST IMAGE TO TILE BUFFER
408
RENDER TILE OF FIRST IMAGE
410
RENDER TILE OF SECOND IMAGE FROM TILE DATA FOR FIRST IMAGE
412
OBTAIN TILE DATA FOR SECOND IMAGE FROM COMMAND BUFFER IF TILE BUFFER MISS
414
ADDITIONAL TILES FOR FIRST AND SECOND IMAGE?
NO
YES
416
EXTRACT NEXT TILE DATA FOR FIRST IMAGE FROM SCENE DATA AND WRITE NEXT TILE DATA TO TILE BUFFER
418
SEND FIRST IMAGE TO FIRST EYE DISPLAY
420
SEND SECOND IMAGE TO SECOND EYE DISPLAY
END

FIG. 4

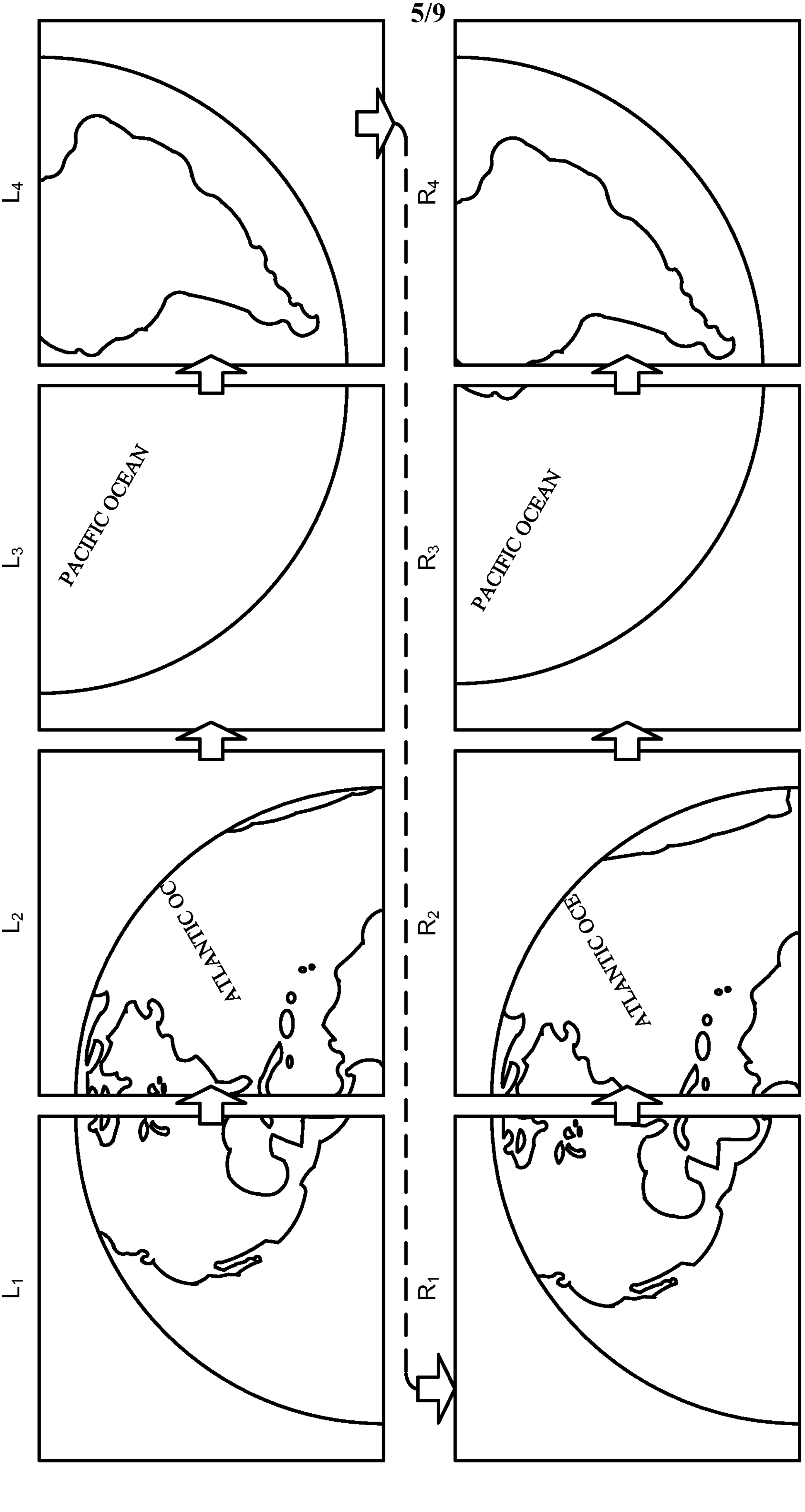
L_4
L_3
PACIFIC OCEAN
L_2
ATLANTIC OC
L_1
R_4
R_3
PACIFIC OCEAN
R_2
ATLANTIC OCE
R_1

FIG. 5A

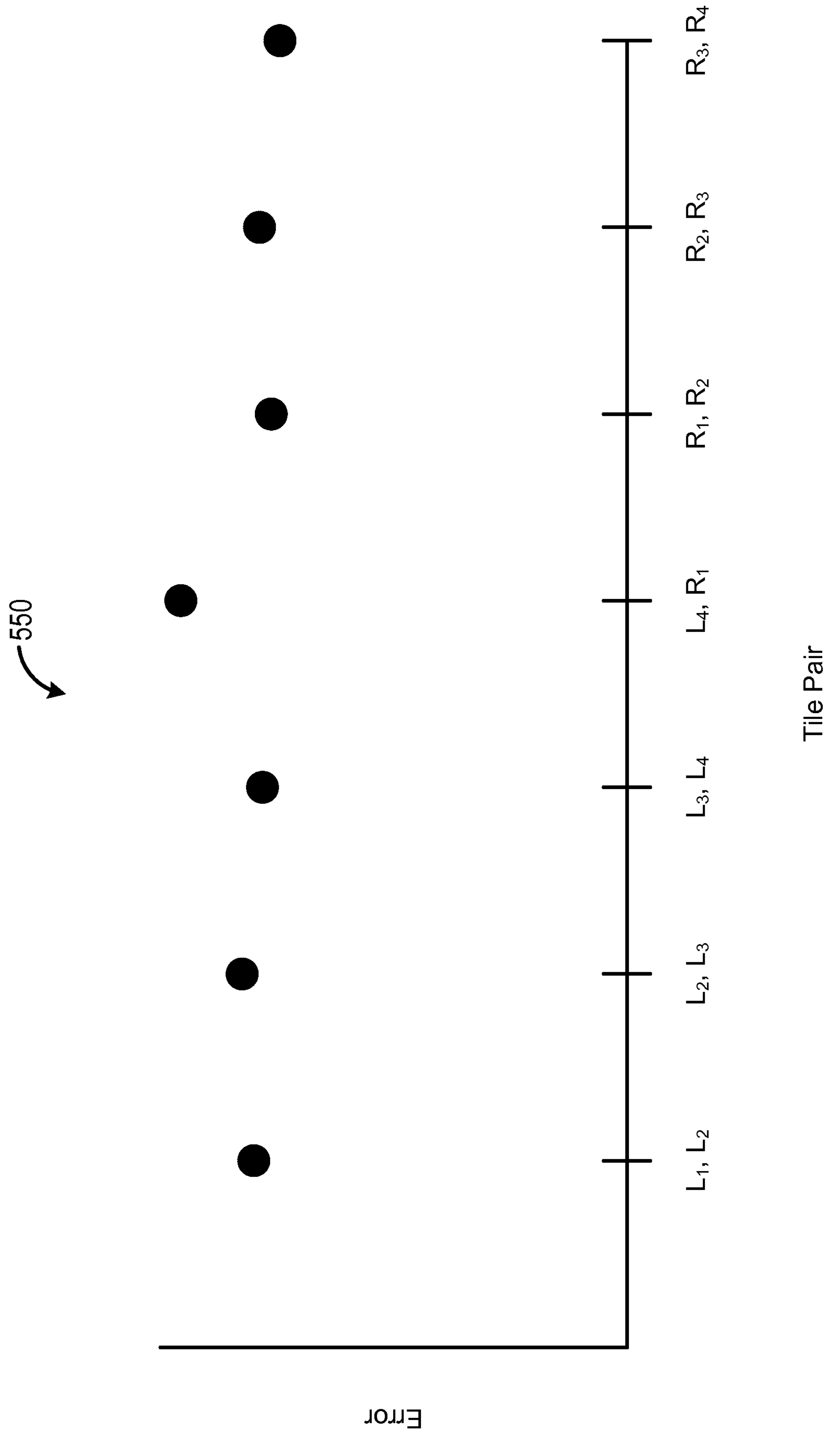

FIG. 5B

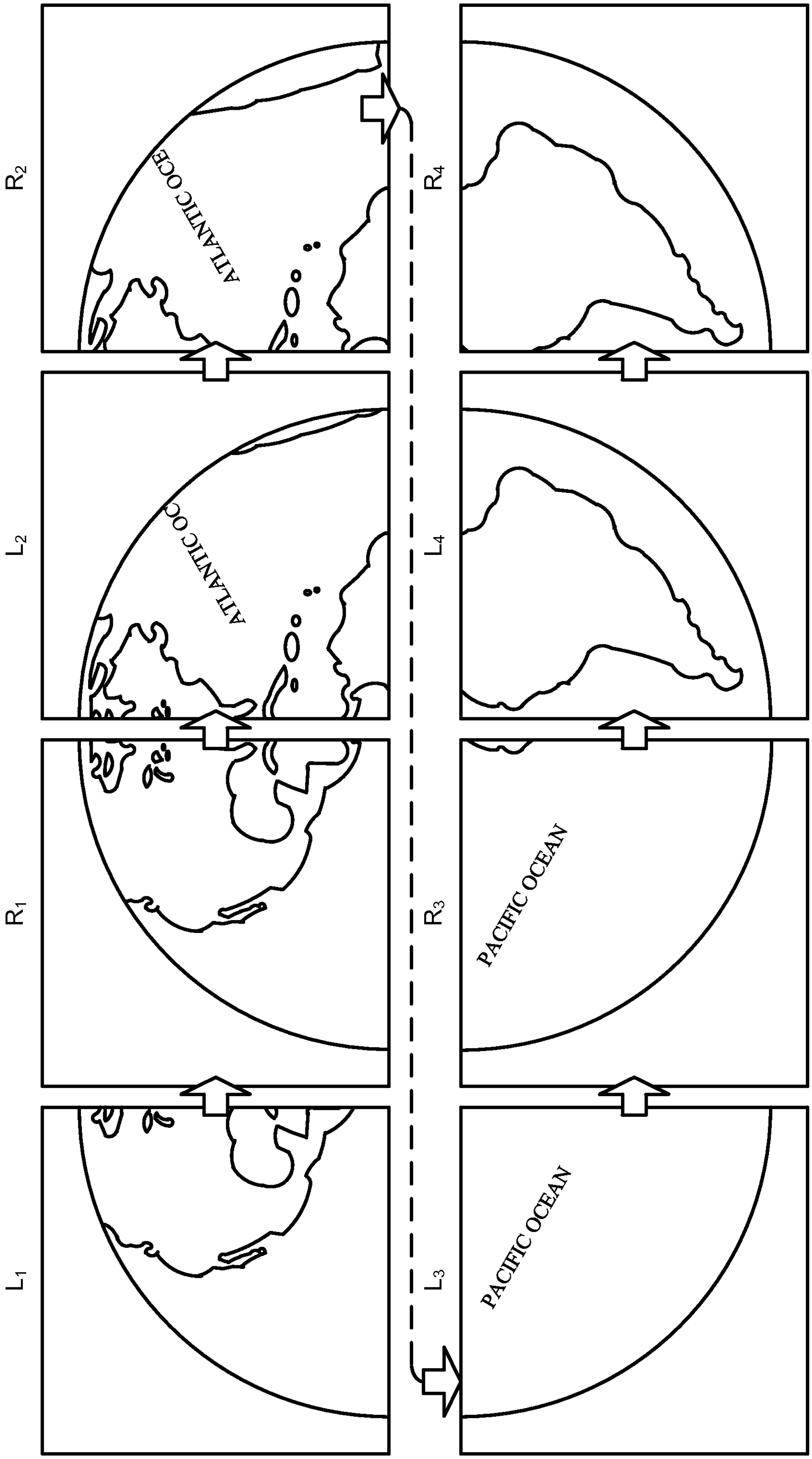
L_1
R_1
L_2
R_2
L_3
R_3
L_4
R_4
ATLANTIC OCE
ATLANTIC OC
PACIFIC OCEAN
PACIFIC OCEAN

FIG. 6A

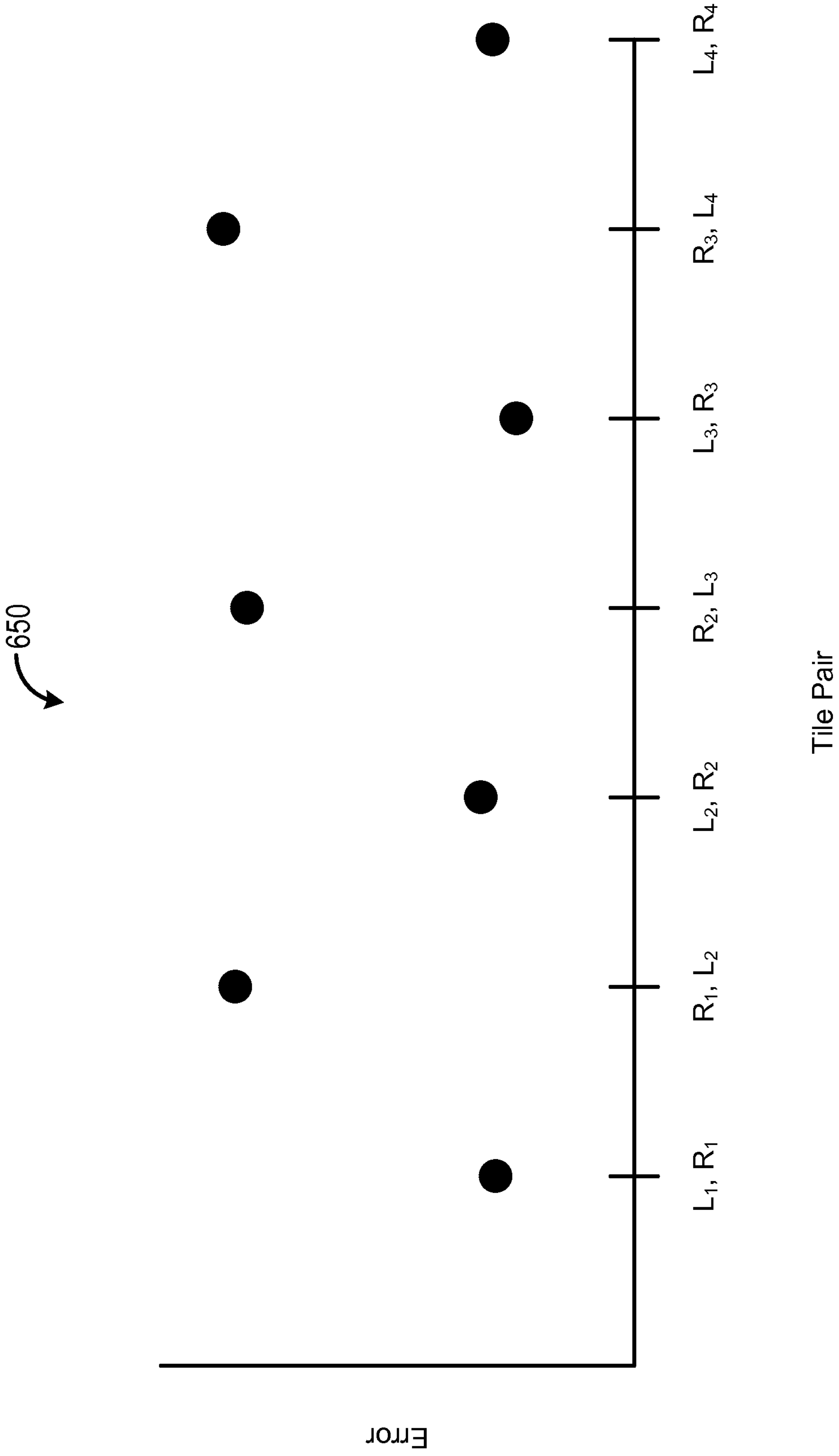

FIG. 6B

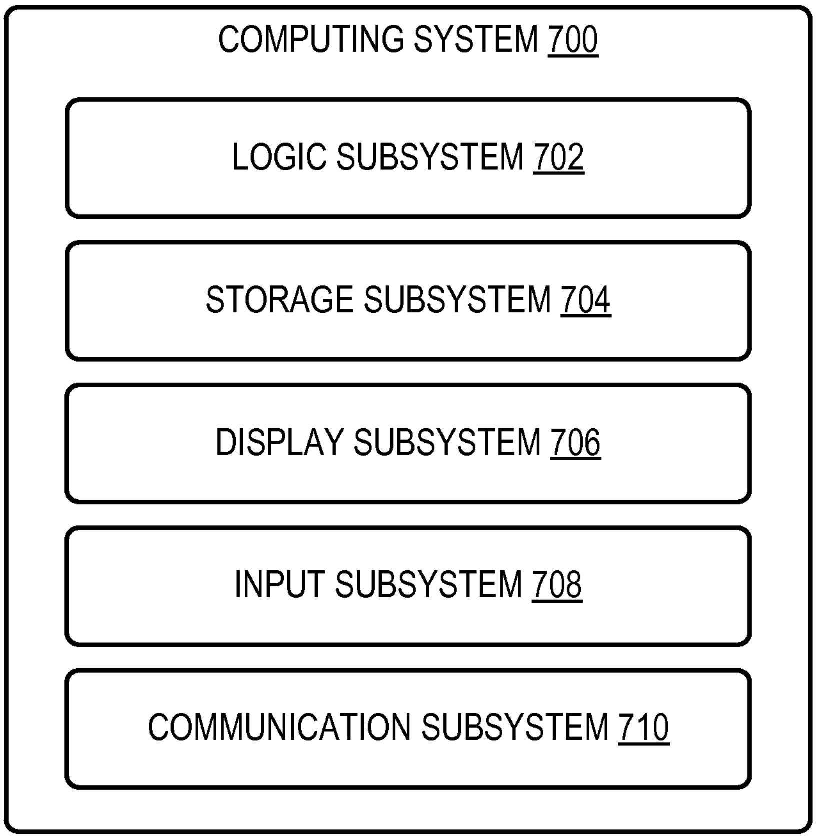
COMPUTING SYSTEM 700
LOGIC SUBSYSTEM 702
STORAGE SUBSYSTEM 704
DISPLAY SUBSYSTEM 706
INPUT SUBSYSTEM 708
COMMUNICATION SUBSYSTEM 710

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/043302

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N13/02 G06T1/60 G06T15/00
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2013/113783 A1 (POURBIGHARAZ FARIBORZ [CA]) 9 May 2013 (2013-05-09) paragraph [0072] - paragraph [0075] -----	1-10
A	US 2012/236002 A1 (BI NING [US] ET AL) 20 September 2012 (2012-09-20) paragraph [0030] -----	1-10
A	US 2011/063285 A1 (HOOVER RUSSELL DEAN [US] ET AL) 17 March 2011 (2011-03-17) the whole document -----	1-10
A	US 6 870 539 B1 (BOWER K SCOTT [US] ET AL) 22 March 2005 (2005-03-22) paragraph [0116]; figures 15,16 -----	1-10

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

15 October 2014

Date of mailing of the international search report

13/11/2014

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

Authorized officer

Montanari, Marco

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2014/043302

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013113783 A1	09-05-2013	CN 103947195 A	23-07-2014
		EP 2777289 A1	17-09-2014
		US 2013113783 A1	09-05-2013
		WO 2013070331 A1	16-05-2013
US 2012236002 A1	20-09-2012	CN 103444190 A	11-12-2013
		CN 103493102 A	01-01-2014
		EP 2686833 A1	22-01-2014
		EP 2687018 A1	22-01-2014
		JP 2014512032 A	19-05-2014
		JP 2014516427 A	10-07-2014
		KR 20130131480 A	03-12-2013
		KR 20130136536 A	12-12-2013
		US 2012235999 A1	20-09-2012
		US 2012236002 A1	20-09-2012
		WO 2012125545 A1	20-09-2012
		WO 2012125648 A1	20-09-2012
US 2011063285 A1	17-03-2011	NONE	
US 6870539 B1	22-03-2005	NONE	