July 2023

# ARTIFICIAL REALITY INTERACTION MODELS

# ARTIFICIAL REALITY INTERACTION MODELS

## BACKGROUND

**[0001]** Artificial reality (XR) devices such as head-mounted displays (e.g., smart glasses, VR/AR headsets), mobile devices (e.g., smartphones, tablets), projection systems, “cave” systems, or other computing systems can present an artificial reality environment where users can interact with "virtual objects" (i.e., computer-generated object representations) appearing in an artificial reality environment. These artificial reality systems can track user movements and translate them into interactions with the virtual objects. For example, an artificial reality system can track a user's hands, translating a grab gesture as picking up a virtual object.

**[0002]** Artificial reality (XR) devices are becoming more prevalent. As they become more popular, the applications implemented on such devices are becoming more sophisticated. Augmented reality (AR) applications can provide interactive 3D experiences that combine images of the real-world with virtual objects, while virtual reality (VR) applications can provide an entirely self-contained 3D computer environment. For example, an AR application can be used to superimpose virtual objects over a video feed of a real scene that is observed by a camera. A real-world user in the scene can then make gestures captured by the camera that can provide interactivity between the real-world user and the virtual objects. Mixed reality (MR) systems can allow light to enter a user's eye that is partially generated by a computing system and partially includes light reflected off objects in the real-world. AR, MR, and VR experiences can be observed by a user through a head-mounted display (HMD), such as glasses or a headset.

**[0003]** Artificial reality systems have grown in popularity with users, and this growth is predicted to accelerate. Some artificial reality environments include visual augments, such as a virtual object or an overlay. In an augmented reality or mixed reality environment, the artificial reality system can add augments that overly or are proximate to real-world objects, for example to enhance the visual appearance of the objects, add an interactive component to the objects, and/or provide a user additional information about the objects.

Given that XR systems include constrained computing resources, systems that efficiently generate augments can provide substantial value.

## SUMMARY

**[0004]** Aspects of the present disclosure are directed to selection disambiguation through zoom and gaze on an artificial reality (XR) device. The technology can render a dense layout of interactive mechanisms (e.g., selectable text and/or graphics) that are responsive to low accuracy input methods on the XR device. Implementations can use a camera to detect a selection gesture (e.g., a pinch) by a user of the XR device. In response to the selection gesture, implementations can render a zoomed-in view displaying a density of interactive mechanisms matching the accuracy of the input method. Implementations can capture a gaze direction of the user of the XR device (e.g., using a camera directed at the user's eyes), and can determine whether the gaze direction is toward one of the interactive mechanisms displayed in the zoomed-in view. If the gaze direction is toward an interactive mechanism, implementations can select that interactive mechanism and take an appropriate action, such as popping up additional content associated with the interactive mechanism.

**[0005]** Further aspects of the present disclosure are directed to creating and delivering shortcuts associated with real-world objects in an artificial reality (XR) environment. Using an XR device, a user can associate a shortcut with the physical object (e.g., an action relative to the physical object, an option to perform an action relative to the physical object, etc.). The shortcut can be anchored to the physical object in XR and delivered when the object anchor is in view of the XR device, and, in some implementations, based on other conditions as well (e.g., time, who else is present, etc.). When the shortcut is an option to perform an action relative to the physical object, the user can use the shortcut by simply tapping a virtual button corresponding to the shortcut with his hand. Similar to creating a shortcut, a user can further remove and change existing shortcuts.

**[0006]** Yet Further aspects of the present disclosure are directed to artificial reality (XR) augmentation using cascading models. Implementations include initial stage model(s) and next stage model(s). In some cases, the include initial stage model(s) are on-device models and next stage model(s) are remotely located model(s). A workload manager can select an augmentation workload using one or more of captured environment data (e.g., captured visual frames, audio, etc.), output from the initial stage model(s), any other suitable data. Implementations of the XR system can process the environment data via the initial stage model(s) and select an augmentation workload. Data for the augmentation workload can then be provided for performing the selected augmentation workload on the data to generate augmentation data via the next stage model(s), and returns the augmentation data to the XR system. The augmentation data can then be output by the XR system to a user.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0007]** Figure 1A is a conceptual diagram illustrating an exemplary display of a zoomed-out view of content on an artificial reality device according to some implementations of the present technology.

**[0008]** Figure 1B is a conceptual diagram illustrating an exemplary display of a zoomed-out view of content on an artificial reality device with a user making a selection gesture to select an interactive mechanism in the view according to some implementations of the present technology.

**[0009]** Figure 1C is a conceptual diagram illustrating an exemplary display of a zoomed-in view of content on an artificial reality device in response to a selection gesture being detected according to some implementations of the present technology.

**[0010]** Figure 1D is a conceptual diagram illustrating an example of a gaze direction of a user overlaid on a display of a zoomed-in view of content on an artificial reality device according to some implementations of the present technology.

**[0011]** Figure 1E is a conceptual diagram illustrating an exemplary display of a view of content on an artificial reality device in response to detecting a gaze direction of a user toward an interactive mechanism according to some implementations of the present technology.

**[0012]** Figure 2 is a flow diagram illustrating a process used in some implementations for selection disambiguation through zoom and gaze on an artificial reality device according to some implementations of the present technology.

**[0013]** Figure 3A is a conceptual diagram of an example view on an artificial reality device of a shortcut being displayed in response to recognizing a photograph in a real-world environment.

**[0014]** Figure 3B is a conceptual diagram of an example view on an artificial reality device of a message being displayed, in conjunction with a recognized photograph, responsive to selection of a shortcut on another artificial reality device.

**[0015]** Figure 4 is a conceptual diagram of an example view on an artificial reality device of a timer being displayed in response to recognizing a spoon in a real-world environment.

**[0016]** Figure 5A is a conceptual diagram of an example view on an artificial reality device of recognized objects having associated shortcuts in a real-world environment, such as a smart home.

**[0017]** Figure 5B is a conceptual diagram of an example view on an artificial reality device of a virtual menu of shortcuts available with respect to a floor lamp in a real-world environment, such as a smart home.

**[0018]** Figure 5C is a conceptual diagram of an example view on an artificial reality device of a floor lamp being turned on in a real-world environment, in response to a selection of a shortcut from a virtual menu.

**[0019]** Figure 6 is a flow diagram illustrating a process used in some implementations for delivering a shortcut associated with a real-world object in an artificial reality environment.

**[0020]** Figure 7 is a conceptual diagram of artificial reality (XR) augmentation architecture using cascading models.

**[0021]** Figure 8 illustrates artificial reality (XR) displays with an augmentation overlay.

**[0022]** Figure 9 illustrates an artificial reality (XR) display with an augmentation overlay and user interaction.

**[0023]** Figure 10 is a flow diagram illustrating a process used in some implementations for artificial reality (XR) augmentation using cascading models.

**[0024]** Figure 11 is a block diagram illustrating an overview of devices on which some implementations of the present technology can operate.

**[0025]** Figure 12 is a block diagram illustrating an overview of an environment in which some implementations of the present technology can operate.

## DESCRIPTION

**[0026]** Aspects of the present disclosure are directed to selection disambiguation through zoom and gaze on an artificial reality (XR) device. The technology can render a dense layout of interactive mechanisms (e.g., selectable text and/or graphics) that are responsive to low accuracy input methods on the XR device. Implementations can use a camera to detect a selection (e.g., pinch, tap, swipe, etc.) gesture by a user of the XR device. In response to the selection gesture, implementations can render a zoomed-in view displaying a density of interactive mechanisms matching the accuracy of the input method. Implementations can capture a gaze direction of the user of the XR device (e.g., using a camera directed at the user’s eyes), and determine whether the gaze direction is toward one of the interactive mechanisms displayed in the zoomed-in view. If the gaze direction is toward an interactive mechanism, implementations can select that interactive mechanism and take an appropriate action, such as showing additional content associated with the interactive mechanism, moving to another page, switching artificial reality environment content, etc. An “artificial reality device,” as used herein, refers to any device that can display, render, process, and/or facilitate displaying, rendering, and/or processing of an artificial reality experience.

**[0027]** For example, a user can view a webpage on an XR headset, the webpage having a dense layout of interactive mechanisms, e.g., a textual hyperlink associated with the word “restaurants,” a textual hyperlink associated with the phrase “grocery stores,” a graphical hyperlink image of a hamburger, and a playable video, that are responsive to low accuracy input methods. Because the view of the webpage is zoomed out to give the user as much information as possible, it may be difficult for the XR headset to ascertain at which of the interactive mechanisms the user is looking and intending to select. Thus, the XR headset can detect a selection gesture by the user in a certain area of the view of the webpage, and zoom in on that area, e.g., the area including the textual hyperlink associated with the phrase “grocery stores” and the graphical hyperlink image of a hamburger, to display a density of interactive mechanisms that match the accuracy of the input method. In other words, the XR headset can match the scale of the displayed interactive mechanisms to the input method accuracy to guarantee that the interactive mechanisms displayed in the zoomed-in view are of a size that accuracy of the input method is sufficient. Because the view of the webpage is zoomed in and less interactive mechanisms are shown in the view, the XR headset can more easily ascertain which of the interactive mechanisms the user’s gaze is focused on, e.g., the image of the hamburger. The XR headset can select the image of the hamburger and open further content related to the image of the hamburger, such as recipes for hamburgers, local restaurants serving hamburgers, etc.

**[0028]** Figure 1A is a conceptual diagram illustrating an exemplary display 100A of a zoomed-out view 106A of content on an XR device according to some implementations of the present technology. Display 100A includes zoomed-out view 106A of content, navigation panels 102A-B, and home bar 104. In some implementations, a user of the XR device can interact with navigation panels 102A-B to control zoomed-out view 106A of content that is displayed, such as by navigating forward, navigating backward, reloading, and/or exiting zoomed-out view 106A. In some implementations, a user can interact and/or view home bar 104 to navigate to a main menu outside of the zoomed-out view 106A, to get information about the XR device (e.g., battery level, notifications, etc.), and/or any other relevant information (e.g., time, weather, etc.).

**[0029]** In exemplary display 100A, zoomed-out view 106A can be a webpage including news content, e.g., an article about inflation. Zoomed-out view 106A can include a plurality of interactive mechanisms 108-116; in this case, hyperlinks associated with particular text. Interactive mechanism 108 can be associated with the phrase, “inflation continues to rise”; interactive mechanism 110 can be associated with the word “gasoline”; interactive mechanism 112 can be associated with the phrase “cost-of-living adjustments”; interactive mechanism 114 can be associated with the phrase “mortgage rates”; and interactive mechanism 116 can be associated with the phrase “home prices.” Because view 106A is zoomed-out, and/or because there are multiple interactive mechanisms 108-116 included in zoomed-out view 106A, it may be difficult for the XR device to ascertain which one of the interactive mechanisms, of the plurality of interactive mechanisms 108-116, at which a selection gesture is directed.

**[0030]** Figure 1B is a conceptual diagram illustrating an exemplary display 100B of a zoomed-out view 106A of content on an XR device with a user making a selection gesture 118 (in this case a pinch gesture) in order to select an interactive mechanism in the view 106A according to some implementations of the present technology. As described further herein, some implementations can capture a single or multiple images representative of a selection gesture by the user using any suitable device internal or external to the XR device, such as an XR headset. For example, images can be captured by one or more image capture devices (e.g., one or more cameras) integral with an XR headset and pointed away from the user, and/or one or more image captures device separate from and proximate to the XR headset and pointed toward the user. As described further herein, some implementations can analyze the image(s) and identify a selection gesture using object recognition techniques and/or a machine learning model trained on known image(s) of selection gestures. In some implementations, selection gesture 118 can be made outside the field of view of one or more image capture devices (and/or without the presence or one or more images capture devices), and can instead be detected using an electromyography (EMG) wearable device on the finger(s) and/or hand(s), such as an EMG bracelet or wristband, an EMG ring, etc., in communication with the XR device.

**[0031]** In some implementations, selection gesture 118 can be in a particular location with respect to zoomed-out view 106A, indicating that the user wishes to select an interactive mechanism in on a particular portion of the content. For example, in Figure 1B, selection gesture 118 can be made toward the top of zoomed-out view 106A, indicating the user wishes to select an interactive mechanism in the top portion of the content. In some implementations, selection gesture 118 can be made either outside or inside the field of view of the user on the content, and the user can indicate where she wishes to select an interactive mechanism based on her gaze direction on the content. Thus, for example, selection gesture 118 can be made anywhere within or outside of zoomed-out view 106A, but her gaze can be pointed toward the top of zoomed-out view 106A, indicating that the user wishes to select an interactive mechanism in the top portion of the content.

**[0032]** Figure 1C is a conceptual diagram illustrating an exemplary display 100C of a zoomed-in view 106C of content on an XR device in response to a selection gesture being detected according to some implementations of the present technology. Because selection gesture 118 was made toward the top of zoomed-out view 106A, zoomed-in view 106C can include content from the top of zoomed-out view 106A, e.g., the title and first couple sentences of content. Zoomed-in view 106C can include a subset of the plurality of interactive mechanisms 108-116. For example, zoomed-in view 106C can include interactive mechanism 108 associated with the phrase, “inflation continues to rise” and interactive mechanism 110 associated with the word “gasoline”; etc. Because view 106C is zoomed-in, and/or because there are less interactive mechanisms 108-112 included in zoomed-in view 106B, it may be easier for the XR device to ascertain a gaze direction of a user of the XR device toward a particular interactive mechanism of the subset of interactive mechanisms 108-112.

**[0033]** Figure 1D is a conceptual diagram illustrating an example of a gaze direction 120 of a user overlaid on a display 100D of a zoomed-in view 106C of content on an XR device according to some implementations of the present technology. As described further herein, some implementations can capture the gaze direction 120 of the user using a camera or other image capture device integral with or proximate to the XR device within image capture range of the user. Images from the camera can be used by a machine learning model to estimate an eye position within the user's head. Some implementations can model and map the eye position of the user relative to the world to determine a vector representing the user's gaze direction 120 through the XR device (e.g., an XR headset).

**[0034]** As described further herein, some implementations can determine whether the gaze direction 120 is directed at a location assigned to an interactive mechanism of interactive mechanisms 108-112 included in the zoomed-in view 106C of the content. Some implementations can make this determination by detecting the direction of the eyes of the user relative to the virtual location of the interactive mechanisms 108-112. As shown in Figure 1D, gaze direction 120 of the user is toward interactive mechanism 108 associated with the phrase "inflation continues to rise."

**[0035]** It is contemplated that in some implementations, gaze direction 120 can include more than one interactive mechanism of interactive mechanisms 108-112, and/or gaze direction 120 still cannot be determined from zoomed-in view 106C. Thus, some implementations can render a further zoomed-in view of the content (not shown) in iterations until gaze direction 120 can be determined and/or until gaze direction 120 includes only one interactive mechanism of interactive mechanisms 108-112.

**[0036]** Figure 1E is a conceptual diagram illustrating an exemplary display 100E of a view 106E of content on an XR device in response to detecting a gaze direction 120 of a user toward an interactive mechanism 108 according to some implementations of the present technology. Because gaze direction 120 of the user is toward interactive mechanism 108, some implementations can select interactive mechanism 108. Selection of interactive mechanism 108 can expand further content related to interactive mechanism 108; in this example, a pop-up 122 including a graph showing rising inflation month over month.

**[0037]** Figure 2 is a flow diagram illustrating a process 200 used in some implementations for selection disambiguation through zoom and gaze on an XR device. In some implementations, process 200 can be performed as a response to a user request to render a view of content. In some implementations, process 200 can be performed by an

XR device displaying or facilitating display or rendering of the view of content, such an XR headset, particular components of the XR headset, and/or one or more XR processing devices in operable communication with the XR headset. In some implementations, portions of process 200 can be performed by one or more components of the XR headset, while other portions of process 200 can be performed by another XR device in operable communication with the XR headset. In some implementations, process 200 can be performed by a server located remotely from the XR device. In some implementations, some blocks of process 200 can be performed by the XR device, while other blocks can be performed by the remote server.

**[0038]** At block 202, process 200 can render a zoomed-out view of content having a plurality of interactive mechanisms on the XR device. Each of the plurality of interactive mechanisms can be graphical, textual, or a combination thereof. For example, the interactive mechanisms can be text, textual or graphical hyperlinks, images, audio playback mechanisms, videos, video playback mechanisms, buttons, input fields, check boxes, toggle switches, radio buttons, sliders, list boxes, dropdown lists, icons, scrollbars, virtual objects, and/or any other textual or graphical control element or combination thereof, such as an image having selectable text or multiple selectable text options.

**[0039]** At block 204, process 200 can detect a selection gesture by a user of the XR device indicative of an intent to select an interactive mechanism in a particular portion of the content. Process 200 can capture multiple images indicative of motion and/or capture a single image representative of gestures by the user using any suitable device internal or external to the XR device (e.g., an XR headset). For example, images can be captured by one or more image capture devices (e.g., one or more cameras) integral with an XR headset and pointed away from the user, or separate from and proximate to the XR headset and pointed toward the user.

**[0040]** Process 200 can analyze the image(s) and identify a selection gesture. For example, when the image(s) are captured by an image capture device, process 200 can perform object recognition on the captured image(s) to identify a user’s hand, and determine that the user’s fingers are closing in a manner representative of a pinch gesture, that the user's finger is extended and performs a motion representative of a tap gesture, etc. In some implementations, process 200 can train a machine learning model with images capturing known selection gestures, such as images showing a user's hand or particular fingers being open then closing, or being closed in a manner indicating of a selection gesture. Process 200 can identify relevant features in the images, such as edges, curves, colors, etc., indicative of fingers and/or a hand. Process 200 can train the machine learning model using these relevant features of known selection gestures. Once the model is trained with sufficient data, process 200 can use the trained model to identify relevant features in newly captured image(s) and compare them to the features of known selection gestures. In some implementations, process 200 can use the trained model to assign a match score to the newly captured image(s), e.g., 75%. If the match score if above a threshold, e.g., 70%, process 200 can classify motion data captured in the images, and/or a hand position in a single image, as being indicative of a selection gesture. In some implementations, process 200 can further receive feedback from the user regarding whether the identification of the selection gesture was correct, and update the trained model accordingly. In some implementations, process 200 can detect the selection gesture using an EMG wearable device by analyzing electrical signals running through the user's arm muscles to translate finger and/or hand movements into the selection gesture.

**[0041]** It is contemplated that process 200 can identify any suitable gesture that can be associated with or indicative of an intention to select an interactive mechanism in a particular portion of the content, thus causing process 200 to zoom in on that portion of the content. For example, process 200 can identify a pinch gesture, a tap gesture, a pointing gesture, a circling gesture, an underlining gesture, etc., which in some implementations can be on a virtual touchscreen. In some implementations, process 200 can alternatively or additionally receive input associated with or indicative of an intention to select an interactive mechanism in content from an input device, such as one or more handheld controllers that allow the user to interact with the view of the content presented by an XR headset. The controllers can include various buttons and/or joysticks that a user can actuate to provide selection input and interact with the content.

**[0042]** At block 206, process 200 can render a zoomed-in view of the content in response to detecting the selection gesture, where the zoomed-in view can include at least one interactive mechanism of the plurality of interactive mechanisms. In some implementations, process 200 can render the zoomed-in view of the content based on the placement of the user's hand when making the selection gesture. For example, process 200 can capture image(s) of the user's hand making a selection gesture and determine the position of the selection gesture relative to the virtual view of the content, e.g., at a particular position on a webpage, toward particular virtual object(s), etc. Process 200 can then render the zoomed-in view at that position. In some implementations, process 200 can render the zoomed-in view of the content based on where the gaze of the user is focused regardless of where the selection gesture is detected. In some implementations, the zoomed-in view can include only one or a subset of the plurality of interactive mechanisms, such that less interactive mechanisms are present in the view and can be selected. In some implementations, the zoomed in view can include all of the plurality of interactive mechanisms, although the plurality of interactive mechanisms can be better differentiated in the zoomed-in view because they are larger.

**[0043]** In some implementations, process 200 can render the zoomed-in view of the content in response to detecting the selection gesture, regardless of whether process 200 understands which interactive mechanism the user intended to select with the selection gesture from the zoomed-out view. In other implementations, process 200 can render the zoomed-in view of the content in response to detecting the selection gesture only when process 200 is unsure which interactive mechanism the user intended to select with the selection gesture from the zoomed-out view. In other words, in the latter implementations, process 200 can be performed without block 206 and/or blocks 208, 210, and automatically select the interactive mechanism on which the selection gesture is made, and/or select the interactive mechanism on which the selection gesture is made and the user's gaze direction is toward, as described further herein.

**[0044]** At block 208, process 200 can capture a gaze direction of the user of the XR device. The gaze direction can be toward an interactive mechanism of the at least one interactive mechanism displayed in the zoomed-in view. Process 200 can capture the gaze direction of the user using a camera or other image capture device integral with or proximate to the XR device within image capture range of the user. For example, process 200 can apply a light source directed to the user's eye which causes multiple reflections around the cornea that can be captured by a camera also directed at the eye. Images from the camera can be used by a machine learning model to estimate an eye position within the user's head. In some implementations, process 200 can also track the position of the user's head, e.g., using cameras that track the relative position of an XR headset with respect to the world, and/or one or more sensors of an inertial measurement unit (IMU) in an XR headset, such as a gyroscope and/or compass. Process 200 can then model and map the eye position and head position of the user relative to the world to determine a vector representing the user's gaze through the XR headset.

**[0045]** Process 200 can determine whether the gaze direction is directed at a location assigned to an interactive mechanism of the at least one interactive mechanism included in the zoomed-in view of the content. Process 200 can make this determination by detecting the direction of the eyes of the user relative to the virtual location of the interactive mechanisms. For example, process 200 can determine if the vector gaze direction passes through an area of the XR device's display showing an interactive mechanism and/or can compute a distance between the point the vector gaze direction passes through the XR device's display and the closest point on the display showing the interactive mechanism. If the gaze direction is directed at the location assigned to an interactive mechanism, process 200 can proceed to block 210.

**[0046]** At block 210, process 200 can select the interactive mechanism of the at least one interactive mechanism that the user's gaze direction is toward. In some implementations, selection of the interactive mechanism can expand further content related to the interactive mechanism. For example, selection of a hyperlink associated with the phrase "hiking trails in Virginia" can cause process 200 to load a pop-up including a list of hiking trails in Virginia. In another example, selection of an avatar of a user in a virtual world can cause process 200 to load a pop-up including details about that user, such as a username, an experience level, etc. In some implementations, selection of the interactive mechanism can redirect the user to another virtual location. For example, selection of a hyperlink on a webpage can cause process 200 to load a view of another webpage. In another example, selection of a virtual object within a virtual world, such as a cave, a football, etc., can cause process 200 to load a view of another virtual world.

**[0047]** Aspects of the present disclosure are directed to creating and delivering shortcuts associated with real-world objects in an artificial reality (XR) environment. A user can don or activate his XR device (e.g., a head-mounted display (HMD) or headset, used interchangeably herein) and view his real-world environment, possibly with virtual objects overlaid thereon, such as in an augmented reality (AR) or mixed reality (MR) experience. In some implementations, the user can create a shortcut associated with a physical object in the real-world environment by selecting the physical object (e.g., by pointing at the physical object, outlining the physical object with his hand, touching the physical object with a controller, etc.), and selecting a shortcut (e.g., an action, an option to perform an action, etc.) from a list of predefined actions that can be performed relative to the physical object. In some implementations, the shortcut can be tied to a particular instance of a physical object (e.g., a particular cup with a particular pattern).

**[0048]** Some implementations can anchor the shortcut to the physical object and deliver the shortcut when the object anchor is in view of the XR device, e.g., as determined by capturing images of the real-world environment and performing object recognition and/or object instance recognition on the physical object. To perform an action associated with the shortcut, the user can simply tap a virtual button corresponding to the shortcut, select a physical button (e.g., on a controller in operable communication with the XR device), and/or audibly announce that the action should be performed. The user can further remove and change existing shortcuts, similar to how shortcuts are created (e.g., by selecting a physical object and selecting to remove and/or modify an existing shortcut).

**[0049]** For example, a user can register a shortcut to call a friend to a real-world gift that the friend gave her for the holidays (e.g., a snow globe) using her XR headset. Thus, when the snow globe comes into view of the XR headset, the XR headset (or other components of an XR system in operable communication of the XR headset) can perform object recognition to identify the snow globe and/or object instance recognition to identify the particular snow globe that the friend gave her, having certain features unique from other snow globes that the user has. The XR headset (or other components of the XR system in operable communication with the XR headset) can retrieve the registered shortcut associated with the snow globe (and/or that instance of the snow globe), and display a virtual button with an option to call the friend.

**[0050]** Thus, implementations can help users achieve more in their lives based on the context, objects, and world around them using the shortcuts described herein. Such shortcuts can not only control existing smart devices (e.g., Internet of Things devices, devices that are connected to a network, such as WiFi or Bluetooth, etc.), but can make even "dumb" devices have some additional functionality. Further, some implementations can not only recognize types of physical objects, but can also recognize particular physical objects (i.e., different instances of physical objects), such that the user experience is more personalized. In addition, in some implementations, the shortcuts can be limited to a particular XR device in a particular real-world environment (e.g., virtual notes created with respect to physical objects can only be displayed to the creator of those virtual notes). However, in some limitations, some shortcuts can be persistent across multiple XR devices in the same real-world environment (e.g., by sharing the shortcuts across XR devices and/or by storing the shortcuts on the cloud, and allowing access by particular XR devices associated with particular users). For example, a family can register a shortcut to a virtual calendar with respect to a real-world refrigerator, such that anyone in the household can use their XR device to view, update, modify, etc., the virtual calendar when the refrigerator is in view.

**[0051]** Figure 3A is a conceptual diagram of an example view 300A on an XR device of a shortcut 304 being displayed in response to recognizing a photograph 302 in a real-world environment 308. The XR device can be an augmented reality (AR) and/or mixed reality (MR) device, such that virtual objects (e.g., shortcut 304) can be overlaid onto real-world environment 308. A daughter (having hand 306), who can be wearing the XR device, can register shortcut 304 to an object anchor associated with real-world photograph 302. Shortcut 304 can be, for example, an option to send a message to her mother, who is depicted in photograph 302. Thus, when photograph 302 comes into view of the XR device (as recognized by object recognition and/or object instance recognition techniques, described further herein), the XR device can display shortcut 304. To send a message to her mother, the daughter can simply use her hand 306 to tap (e.g., make a tapping or touching gesture toward) shortcut 304 displayed on the XR device.

**[0052]** Figure 3B is a conceptual diagram of an example view 300B on an XR device of a message 312 being displayed, in conjunction with a recognized photograph 310, responsive to selection of a shortcut 304 on another XR device. For example, responsive to selection of shortcut 304 on the daughter’s XR device, the daughter can generate and send message 312 to her mother’s XR device. The mother can have previously registered a shortcut to display message 312 in conjunction with photograph 310 of her daughter when messages are received from her daughter. Thus, when the daughter sends message 312, message 312 can be displayed on the XR device of the mother when photograph 310 is in view of the XR device (as determined by applying object recognition and/or object instance recognition techniques, as described further herein).

**[0053]** Figure 4 is a conceptual diagram of an example view 400 on an XR device of a timer 404 being displayed in response to recognizing a spoon 402 in a real-world environment 408. The XR device can be an AR and/or MR device, such that virtual objects (e.g., timer 404) can be overlaid onto real-world environment 408. A user can have previously registered a shortcut to an object anchor associated with real-world spoon 402, e.g., timer 404 reminding the user to stir pot 406 after an elapsed period of time. The XR device can perform object recognition and/or object instance recognition techniques (described further herein) to recognize spoon 402 in real-world environment 408, and display a countdown on timer 404 indicating the remaining time until pot 406 needs to be stirred.

**[0054]** Figure 5A is a conceptual diagram of an example view 500A on an XR device of recognized objects having associated shortcuts in a real-world environment 502, such as a smart home. The XR device can be an AR and/or MR device, such that virtual objects (e.g., bounding boxes 504, 508, 512) can be overlaid onto real-world environment 502. When a user of the XR device enters real-world environment 502, the XR device can perform object recognition and/or object instance recognition to recognize physical objects within the real-world environment, and determine which physical object(s) have object anchor(s) with associated shortcut(s). In real-world environment 502, the XR device can recognize thermostat 506, table lamp 510, and floor lamp 514, and indicate that those objects have associated shortcuts by identifying them (e.g., with textual labels) and surrounding them with virtual bounding boxes 504, 508, 512, respectively. In some implementations, thermostat 506, table lamp 510, and floor lamp 514 can be Internet of Things (IoT) or smart devices, and/or can be connected to smart devices (e.g., via a smart wall plug).

**[0055]** Figure 5B is a conceptual diagram of an example view 500B on an XR device of a virtual menu 516 of shortcuts 518, 520 available with respect to a floor lamp 514in a real-world environment, such as a smart home. For example, a user having view 100A can select floor lamp 514, for example, by audibly selecting floor lamp 514 (e.g., “show me options with respect to floor lamp 514,” “turn floor lamp 514 on,” etc.), gesturing toward floor lamp 514 (e.g., pointing at floor lamp 514, as captured by the XR device), or selecting a physical button when a virtual pointer is over floor lamp 514 (e.g., using a controller in operable communication with the XR device). In response to the selection of floor lamp 514, some implementations can display menu 516. Menu 516 can include, for example, a shortcut 518 to turn floor lamp 514 on bright, and a shortcut 520 to turn floor lamp 514 on dim. The user of the XR device can, for example, select shortcut 518 to turn floor lamp 514 on bright, using audible selection, performing a gesture toward shortcut 518 (e.g., pointing), or selecting a physical button when a virtual pointer is over shortcut 518.

**[0056]** Figure 5C is a conceptual diagram of an example view 500C on an XR device of a floor lamp 514 being turned on in a real-world environment 502, in response to a selection of a shortcut 518 from a virtual menu 516. In response to a user selection of the shortcut 518 associated with turning floor lamp 514 on bright, the XR device can directly or indirectly communicate with floor lamp 514 (when floor lamp 514 is a smart device) to turn floor lamp 514 on bright, such as through an application installed on the XR device associated with a developer of smart floor lamp 514. When floor lamp 514 is instead connected to an external smart device (e.g., a smart wall outlet or other smart controller), the XR device can communicate with the external smart device to turn floor lamp 514 on bright, as shown in view 500C.

**[0057]** Figure 6 is a flow diagram illustrating a process used in some implementations for delivering a shortcut associated with a real-world object in an XR environment. In some implementations, process 600 can be performed as a response to detection of activation or donning of an XR device. In some implementations, process 600 can further be performed as a response to launching of an application on the XR device. In some implementations, process 600 can be performed by the XR device, such as an HMD or headset. In some implementations, some or all of the steps of process 600 can be performed by other components in an XR system including the XR device, such as external processing component(s) in operable communication with the XR device. In some implementations, some or all of the steps of process 600 can be performed by a remote server, such as a developer computing system or platform computing system.

**[0058]** At block 602, process 600 can detect activation or donning of an XR device by a user. The XR device can be configured to display at least one of an augmented reality (AR) experience, a mixed reality (MR) experience, or both, as defined further herein, such that at least a portion of a real-world environment of the user is visible on the XR device. In some implementations, process 600 can detect activation or donning of the XR device automatically, e.g., through one or more sensors of an inertial measurement unit (IMU), through temperature sensors, etc., integral with or in operable communication with the XR device. Alternatively or additionally, process 600 can detect launch of an application associated with an AR or MR experience on the XR device.

**[0059]** At block 604, process 600 can identify a real-world (i.e., physical) object in a real-world environment of the user by performing object recognition and/or object detection on one or more images of the real-world environment. Process 600 can capture the one or more images of the real-world environment using one or more image capture devices (e.g., one or more cameras) integral with or in operable communication with the XR device. Process 600 can perform object recognition and/or object detection via any known techniques, such as feature extraction and classification, machine learning models, deep learning models, template matching, image segmentation and blob analysis, etc. In some implementations, process 600 can alternatively or additionally perform object detection on one or more images of the real-world environment to identify the real-world object. In some implementations, process 600 can perform object recognition and/or object detection to identify particular instances of real-world objects, such as a particular coffee mug, instead of identifying a coffee mug in general.

**[0060]** The real-world object can have a corresponding object anchor registered on the XR device. In some implementations, the object anchor can be registered to the real-world object previously by a shortcut creation process. In some implementations, the object anchor can be registered to the real-world object manually (e.g., by the user of the XR device). For example, the user can select a real-world object (e.g., by gesturing at the real-world object as captured by one or more cameras on the XR device, by touching or outlining the real-world object with a controller in operable communication with the XR device, etc.), thereby creating an object anchor with respect to the real-world object. The user can then select a corresponding shortcut to associate with the object anchor via the XR device. In some implementations, the user can select the shortcut from a list of predefined shortcuts available generally, or available for that particular object (i.e., that object or that instance of the object).

**[0061]** In some implementations, the object anchor can be registered to the real-world object automatically. For example, process 600 can automatically register an object anchor with a corresponding shortcut to all objects of a particular type (e.g., a timer shortcut registered to an object anchor associated with a real-world oven or stove). In some implementations, process 600 can automatically register an object anchor with a corresponding shortcut based on learned habits of a user of the XR device, e.g., by training and applying a machine learning model. For example, process 600 can observe that the user always turns on the television when particular real-world objects are in view of the XR device, such as physical objects typically appearing in a bedroom (e.g., a pillow, a bed, blankets, a dresser, etc.), and can automatically register one or more object anchors to one or more of those physical objects. Process 600 can register the one or more object anchors along with a corresponding shortcut to display an option to turn on the television and/or can automatically turn on the television. For example, process 600 can turn on the television via communication with an Internet of Things (IoT) network and/or through direct or indirect communication with the smart device controlling the television (e.g., a smart power outlet in which the television is plugged or another smart controller).

**[0062]** In some implementations, process 600 can further register a shortcut to an object anchor associated with a real-world object based on other conditions and/or parameters of the real-world environment. For example, process 600 can specify that the shortcut only be displayed and/or performed at a certain time of day or based on environmental changes or conditions, such as lighting (e.g., natural, ambient, and/or artificial), temperature, movement, etc. In some implementations, process 600 can detect such conditions via the XR device (e.g., using one or more sensors of an inertial measurement unit (IMU), using one or more image capture and/or light sensing devices, using temperatures sensors, etc.). In some implementations, process 600 can ascertain such conditions via communication with one or more other devices, such as smart and/or IoT devices (e.g., smart lamps, smart thermostats, etc.).

**[0063]** At block 606, process 600 can identify the shortcut associated with the real-world object using the object anchor. Based on the previous association of the real-world object with the object anchor, process 600 can locate the shortcut corresponding to the object anchor. In some implementations, process 600 can use a descriptor of the identified real-world object (e.g., coffee mug, my coffee mug, my most frequently used coffee mug, coffee mug with a star on it, etc.) to access a lookup table or database of real-world objects, object anchors, and associated shortcuts for the real-world environment.

**[0064]** At block 608, process 600 can perform at least one of an action on the XR device, an action in the real-world environment, or both, based on the identified shortcut.

Exemplary actions on the XR device can include, for example, display of one or more options to perform one or more tasks associated with the shortcut, display of particular text and/or graphics based on the identified shortcut, launch of a particular application based on the identified shortcut, etc. Exemplary actions in the real-world environment can include, for example, activating an IoT or smart device, deactivating an IoT or smart device, modifying output of an IoT or smart device, etc. In some implementations, process 600 can perform the one or more actions only when any additional conditions associated with the shortcut are met. For example, process 600 can display an option for a user to turn on the lights only when the room is dark (i.e., has low ambient light).

**[0065]** In some implementations, process 600 can modify and/or remove a shortcut associated with an object anchor of a real-world object, similarly to how process 600 can create a shortcut. For example, process 600 can recognize and/or detect a real-world object in the view of an XR device and display one or more shortcuts linked to the real-world object. A user, via the XR device, can select an option to modify and/or remove the shortcut, such as by an audible command (e.g., “change the shortcut for the lamp to a timer,” “remove the existing shortcut,” etc.), a selection of a virtual button via a gesture detected by the XR device, a selection of a physical button (e.g., on a controller, etc.).

**[0066]** Although described herein with respect to particular examples, it is contemplated that any number of shortcuts can be created for any number of tasks associated with any real-world object. For example, for a real-world object of a photograph of someone, a shortcut of a video call option with a preconfigured contact can be displayed for an associated task of starting a video call. In another example, for any suitable object (e.g., a cooking pot, pan, stove, oven, etc.), a shortcut of a timer setup user interface can be displayed for an associated task of setting a timer. In another example, for a real-world object of a clock, a shortcut of multiple virtual clocks for other time zones can be displayed for a task of finding out the time in other time zones. In another example, for a real-world object of a smart device, a shortcut of a custom set of control actions (e.g., turn on, turn off, turn up volume, turn down volume, set security alarm, set temperature, etc.) can be displayed for a task of controlling the smart device.

**[0067]** In another example, for a real-world object of a plant, a shortcut of a preconfigured reminder with a watering schedule (and optionally logging) can be displayed for the task of reminding someone to water a plant. In another example, for any trigger real-world object (e.g., a birthday gift, a graduation certificate, etc.), a shortcut of a one-click user interface button to replay a video memory can be displayed for the task of replaying a video of a good memory (e.g., a birthday party, a graduation ceremony, etc.). In another example, for a real-world object of a Christmas tree or another Christmas decoration, a shortcut of playing a Christmas songs playlist can be activated for the task of playing Christmas music. In another example, for a real-world object of a children's toy, a shortcut of playing a children's songs playlist can be activated for the task of playing children's music. In another example, for a real-world object of a basketball, a shortcut of a messaging application to a preconfigured group chat can be displayed for the task of group chatting with a user's basketball friends.

**[0068]** In addition, shortcuts can be established for multiple domains, such as communication (e.g., start a video call, send a message, etc.), utilities (e.g., show local time in other time zones, set a timer, add a note, etc.), entertainment (e.g., play music, open a favorite media channel, shortcuts to favorite television programs, shortcut to enter a virtual world, etc.), reminders (e.g., shortcut to set a reminder, etc.), notifications (e.g., pull notifications from a certain source, such as a person, a subscribed channel, a delivery service, an application, etc.), smart home (e.g., custom control menu per device, custom control menu per area, such as living room, kitchen, etc.), lifestyle (e.g., shortcuts to food recipes anchored to objects like toaster, stove, oven, microwave, etc.), community (e.g., shortcuts to a book community on certain topics, such as science fiction, shortcuts to a pet community for a certain pet breed, such as Labrador Retriever, etc.), health and fitness (shortcuts to prescription notes, shortcuts to dietary guidelines, shortcuts to training tutorials, show activity and fitness statistics, show activity goals, etc.), gaming (shortcuts to video games, etc.), and/or the like. It is contemplated, however, that shortcuts can be used in many different environments and scenarios for many different real-world objects and tasks; thus, these examples are not intended to be exhaustive.

**[0069]** Aspects of the present disclosure are directed to artificial reality (XR) augmentation using cascading models. Implementations include initial stage model(s) (which in some cases can be one or more on-device models) that execute at a capture system (e.g., XR system), and next stage model(s) (which in some cases can be one or more models that execute at a remote device - e.g., a cloud computing device, an edge computing device, a smart home computing device, laptop, smartphone, or any other suitable computing device remote from the XR system). The cascading model(s) can distribute computing workloads to different computing devices to achieve efficient generation of augmentation data. In other cases, the initial stage model(s) and the next stage model(s) can be on the same system, which may be the XR system or a system remote from the XR system.

**[0070]** In .some implementations, the initial stage models can process captured data (e.g., camera frames, audio data, etc.) and generate a high-level classification, such as an object classification (e.g., dog, book, landmark, etc.). A workload manager can select an augmentation workload using one or more of: captured data (e.g., captured visual frames, audio, etc.); output from the initial stage model(s); any other suitable data; or any combination thereof. Once the augmentation workload is selected, the XR system can transmit next stage data to remote device(s) loaded with next stage model(s). The next stage data can include the captured data, output from initial stage model(s), or any combination thereof.

**[0071]** The remote device(s) can process the received next stage data using loaded next stage model(s) that correspond to the selected augmentation workflow. For example, next stage model(s) can generate, using the next stage data, one or more of: a granular classification for the detected object (e.g., dog breed for an object classified as a dog, book title and author for an object classified as a book, landmark/POI name and location for an object classified as a landmark/monument, artist and title for an object classified as a painting, etc.), a visual augment, an audio augment, or any other suitable augmentation data.

**[0072]** In some implementations, a set of predefined augmentation workloads available for selection at the XR system can be extensible. For example, any suitable third-party entity can provide access to new augmentation resources (e.g., next stage model(s)) at and software (e.g., workload manager logic) for selecting the new augmentation workload. For example, the new augmentation workload can: classify a new object class into a granular classification (e.g., cat breeds, vehicle make, model and year, etc.); generate new visual augments, for example augments for new classes of objects; or provide any other suitable augmentation.

**[0073]** In some implementations, the cascading model(s) can comprise a multi-level pipeline that includes three or more models and/or other processing algorithms. For example, initial stage model(s) can categorize an object as a high-level class or categorize a portion of a captured frame as containing a high-level class, such as a painting or artwork. A workload manager can select a painting augmentation workload based on the classification. The XR system can transmit next stage data (e.g., the high-level class and the portion of the captured frame that contains the painting) to be operated on by the next stage pipeline. One or more next stage model(s) or other algorithms (e.g., one or more machine learning models trained to identify specific paintings, algorithms to identify colors, shapes, paint types, etc.) can identify the artist, title, year and other features for the painting using the next stage data. In some implementations, an additional workload manager can select an additional augmentation workload based on the specific painting identified. For example, one or more rules can be defined that map augmentation workloads to subsets of specific paintings. The additional augmentation workload may generate a visual augment to be displayed over or proximate to the painting at the XR system.

**[0074]** In some implementations, the next stage model(s) may not include a model that corresponds to the additional augmentation workload, however one or more of next stage model(s) (or other algorithms, which may be loaded at a second remote device) may match the additional augmentation workload. In this example, the device executing the next stage model(s) can transmit next stage data (e.g., the portion of the captured frame that includes the painting, a label that defines the specific augmentation to generate, such as virtual object, filter, animation, etc.), and the next stage model(s) can generate the visual augment using the received next stage data. The next stage models can the return this augmented version of the portion of the captured frame (e.g., to the XR system), which can display the visual augment to the user.

**[0075]** Figure 7 is a conceptual diagram of artificial reality (XR) augmentation architecture using cascading models. System 700 includes XR system 702, cloud computing device 704, and other computing device 706. XR system 702 can include data capture layer 708, initial stage model(s) 710, and selection layer 712. Cloud computing device 704 can include next stage model(s) 714 and output data processing layer 716. Other computing device 706 can include next stage model(s) 718 and output data processing layer 720. In various implementations, various of XR system 702, cloud computing device 704, and other computing device 706 can be implemented as separate devices remote from one another or can be implemented as components of the same computing device (e.g., all part of the same XR device or all on a server system).

**[0076]** XR system 702 can be a system worn by a user, such as a head-mounted display. XR system 702 can capture environment data (e.g., camera frames, audio, etc.) proximate to the user via sensors (e.g., cameras, microphones, etc.). Data capture layer 708 can pass portions of the captured environment data to initial stage model(s) 710. Initial stage model(s) 710 can perform an initial workload on the captured environment data. For example, initial model(s) 700 can include computer vision machine learning models (e.g., object detection and/or high level classification models), audio processing machine learning models, models trained to classify semantic embeddings, workload mapping models, any other suitable machine learning models and/or data processing models, or any combination thereof.

**[0077]** In some implementations, the initial stage model(s) 710 can detect objects within the captured environment data. For example, a computer vision model can detect an object within the captured environment data that corresponds to a real-world object. In some implementations, initial stage model(s) 710 can classify the detected object into a high-level class, such as one or more classes from a limited number of high-level object classes (e.g., 20, 50, 100, 200, 500, and the like). Example high-level object classes include a dog, a book, a monument, a building, a person, a couch, a chair, a car, a bus, and other suitable high-level object classes.

**[0078]** Selection layer 712 can select an augmentation workload using: portions of the environment data, output from initial stage model(s) 710, and any other suitable data. In some implementations, selection layer 712 can include a set of predefined augmentation workloads that correspond to workload resources (e.g., next stage model(s) 714 and 716) loaded at remote computing devices, such as cloud computing device 704 or other computing device 706. Selection layer 712 can select one or more of these predefined workloads for the environment data/output from initial stage model(s) 700. For example, the output can be a high-level object class (e.g., dog, book, monument, painting, etc.) and selection layer 712 can select the augmentation workload that corresponds to the high-level object class (e.g., dog breed classifier, book identifier, point of interest identifier, painting identifier, etc.).

**[0079]** Once the augmentation workload is selected, XR system 702 can transmit next stage data to one or more of cloud computing device 704 or other computing device 706. For example, next stage model(s) 714 loaded at cloud computing device 704 may include one or more models that perform the augmentation workload selected by selection layer 712. The next stage data transmitted from XR system 702 to computing device 704 can include one or more of the captured environment data, output from initial stage model(s) 710, or any combination thereof. For example, next stage data can include a portion of a captured camera frame that contains a detected and classified object, a high-level object label classified by one or more first stage model(s), and any other suitable data.

**[0080]** Cloud computing device 704 can process the next stage data received from XR system 702 using one or more next stage model(s) 714 that correspond to the selected augmentation workflow. For example, next stage model(s) 714 can generate, using the next stage data, one or more of: a granular classification for the detected object (e.g., dog breed for an object classified as a dog, book title and author for an object classified as a book, landmark/POI name and location for an object classified as a landmark/monument, artist and title for an object classified as a painting, etc.), a visual augment, an audio augment, or any other suitable augmentation data.

**[0081]** In some implementations, next stage model(s) 714 can be configured to augment visual data, such as captured camera frames. For example, initial stage model(s) 710 may categorize an object that triggers a rule for generating a visual augment (e.g., an object with a QR code or other suitable code, a brand of clothing, furniture, or other suitable object(s), etc.), and the selected augmentation workload may generate such an augment. In this example, second stage model(s) can comprise one or more machine learning models trained/configured to generate an augmented version of captured visual data (e.g., captured frames), such as Generative Adversarial Networks (GANs), encoder/decoder models, and the like

**[0082]** In some implementations, the set of predefined augmentation workloads at selection layer 712 can be extensible. For example, any suitable third-party entity can provide access to new augmentation resources (e.g., next stage model(s)) at a remote computing device and software (e.g., logic at selection layer 712) for selecting the new augmentation workload. For example, the new augmentation workload can: classify a new object class into a granular classification (e.g., cat breeds, vehicle make, model and year, etc.); generate new visual augments, for example augments for new classes of objects; or provide any other suitable augmentation.

**[0083]** In some implementations, the cascading model(s) can comprise a multi-level pipeline that includes three or more models. For example, initial stage model(s) 710 can categorize an object as a high-level class or categorize a portion of a captured frame as containing a high-level class, such as a painting or artwork. Selection layer 712 can select a painting augmentation workload. XR system 702 can transmit next stage data (e.g., the high-level class and the portion of the captured frame that contains the painting) to other computing device 706. One or more of next stage model(s) 718 at other computing device 706 (e.g., one or more machine learning models trained to identify specific paintings) can identify the artist, title, and year for the painting using the next stage data. In some implementations, an additional selection layer at other computing device 706 (not depicted in Figure 7) can select an additional augmentation workload based on the specific painting identified. For example, one or more rules can be defined that map augmentation workloads to subsets of specific paintings (e.g., paintings by one or more specific artists, paintings from a collection, paintings from a given era, or any combination of these). The additional augmentation workload may generate a visual augment to be display over or proximate to the painting at the XR system.

**[0084]** In some implementations, next stage model(s) 718 can include one or more models that generate the visual augment. In this example, the one or more next stage model(s) can take, as input, the portion of the captured frame that includes the painting and output an augmented version of the portion of the captured frame that includes a visual augmentation (e.g., animation, virtual object, filter, etc.). Other computing device 706 can then return this augmented version of the portion of the captured frame to XR system 702, which displays the visual augment to the user.

**[0085]** In another example, next stage model(s) 718 may not include a model that corresponds to the additional augmentation workload, however one or more of next stage model(s) 714 loaded at cloud computing device 704 may match the additional augmentation workload. In this example, other computing device 706 can transmit next stage data to cloud computing device 704 (e.g., the portion of the captured frame that includes the painting, a label that defines the specific augmentation to generate, such as virtual object, filter, animation, etc.), and next stage model(s) 714 can generate the visual augment using the received next stage data. Cloud computing device 704 can the return this augmented version of the portion of the captured frame to XR system 702, which can display the visual augment to the user.

**[0086]** In some implementations, cascading model(s) may perform character recognition (e.g., optical character recognition) and/or language translation. For example, initial stage model(s) 710 may categorize a portion of a captured frame as containing text. Selection layer 712 can select one or more of an optical character recognition augmentation workload and a language translation augmentation workload for the portion of the captured frame. One or more of next stage model(s) 714 and/or 716 may receive next stage data (e.g., the portion of the frame containing the text) and recognize the characters within the text. The recognized characters and the portion of the captured frame containing the text can also be provided to another one of next stage model(s) 714 and/or 716 to perform language translation (e.g., Spanish to English, English to French, etc.).

**[0087]** Figure 8 illustrates artificial reality (XR) displays with an augmentation overlay. Display 802 includes real-world object 806 and interface 808, and display 804 includes real-world object 806 and augment 810. Displays 802 and 804 can comprise mixed reality, augmented reality, or any other suitable pass-through XR displays output to a user via a XR system. Display 802 displays a captured frame that includes real-world object 806.

**[0088]** In some implementations, the XR system can be operating in an exploration mode. For example, a user can interact with interface 808 to trigger the exploration mode, and in exploration mode the XR system can automatically select augmentation workload(s) based on high-level classification(s) for portions of the frames captured by the XR system. In the illustrated example, one or more initial stage model(s) can classify a portion of a captured frame that includes real-world object 808 with a high-level classification, such as a dog classification. Based on the high-level classification, an augmentation workload for identifying dog breeds can be selected and next stage data (e.g., the portion of the captured frame(s) with real-world object 806) can be transmitted to a remote device.

**[0089]** In some implementations, one or more second stage model(s) at the remote device can classify the dog breed using the next stage data and return the classified dog breed to the XR system. The XR system can output display 804 to a user, which displays real-world object 806, the dog object, and augment 808, which is populated with the classified dog breed information received from the remote device. In some implementations, the user can interact with augment 808 and launch additional interface(s) to explore the information about real-world object 806.

**[0090]** Figure 9 illustrates an artificial reality (XR) display with an augmentation overlay and user interaction. Display 902 includes augment 904. In some implementations, display 902 can be generated after a user interacts with display 804 and augment 808 of Figure 8. For example, augment 904 can be used to further explore interactions with real-world object 806, such as connecting to social media, launching webpages, and other suitable interactions.

**[0091]** In some implementations, multiple high-level classes can be detected at multiple portions of the environment data captured at a XR system. For example, the environment surrounding a user wearing a HMD of a XR system may include a book and a painting. A predefined ranking system can prioritize one or more of the portions of the environment data. For example, a painting may be ranked as a higher priority than a book, and thus an augmentation workload for the portion of the captured frame(s) that include the painting can be selected and implemented. Any other suitable ranking can be implemented.

**[0092]** Figure 10 is a flow diagram illustrating a process used in some implementations for artificial reality (XR) augmentation using cascading models. In some implementations, process 1000 can be performed at a XR system in response to initiation of a user interaction mode (e.g., discover mode, exploration mode, etc.). For example, the user interaction mode can be triggered by a user action, automatically by the XR system, in response to object detection by the XR system, in response to any other suitable trigger, or any combination thereof. In some implementations, process 1002 can be performed by a cloud computing device, edge computing device, home computing device, any suitable computing device remote from the XR system, or any combination thereof.

**[0093]** At block 1004, process 1000 can capture environment data of a real-world environment. For example, the environment data can include visual frames (e.g., captured via one or more cameras), audio (e.g., captured via one or more microphones), and any other suitable environment data. In some implementations, the environment data represents a real-world environment proximate to a user that wears a head-mounted display component of a XR system.

**[0094]** At block 1006, process 1000 can analyze the environment data using one or more first stage models. Example first stage models include computer vision machine learning models (e.g., object detection and/or high level classification models), audio processing machine learning models, models trained to classify semantic embeddings, workload mapping models, and any other suitable data processing models. In some implementations, the first stage model(s) can be stored on-device at the XR system. For example, the environment data captured (e.g., via XR system camera(s), microphone(s), etc.) and the environment data processing via the first stage model(s) can be performed on-device at the XR system.

**[0095]** In some implementations, the first stage model(s) can detect objects within the environment data. For example, a computer vision model can detect an object within the captured environment data that corresponds to a real-world object. In some implementations, the first stage model(s) can classify the detected object into a high-level class, such as one or more classes from a limited number of high-level object classes (e.g., 100, 200, 500, and the like). Example high-level object classes include a dog, a book, a monument, a building, a person, a couch, a chair, a car, a bus, a painting, and other suitable high-level object classes.

**[0096]** At block 1008, process 1000 can select an augmentation workload for the environment data. For example, a selector or mapping model may select an augmentation workload using one or more of: the environment data, output from one or more first stage model(s), or any combination thereof. In some implementations, one or more of the first stage models can be configured/trained to select an augmentation workload using the captured environment data. For example, the first stage model processing can include a sequence of first stage models, where the output from initial model(s) is provided as input to the next model(s). For example, initial first stage model(s), such as a computer vision machine learning model, can perform object detection and/or high level classification, such as classifying an object as a dog, book, monument/landmark, or any other suitable object. A next first stage model, such as a mapping model, can map the classified object to an augmentation workload.

**[0097]** In some implementations, the mapping model(s) can map high-level object classifications to one or more augmentation workload(s). For example, an object classified as a dog can be mapped to a dog breed classification augmentation workload. In another example, an object classified as a book can be mapped to a book classification augmentation workload. In another example, an object classified as a landmark or monument can be mapped to a point of interest (POI) augmentation workload.

**[0098]** In some implementations, the mapping model(s) may map a detected/classified object to a visual augmentation workload. For example, one or more high-level classes of objects can be mapped to a visual augmentation workload, such as a workload for generating an animation, a virtual object augment or overlay, or any other suitable visual augment. In some implementations, the augmentation workload can be selected from among a predefined set of augmentation workloads. For example, the mapping model(s) can map any high-level classification to any other suitable predefined augmentation workload.

**[0099]** At block 1010, process 1000 can determine whether the selected augmentation workload comprises a remote augmentation workload. For example, a first set of augmentation workloads can be performed on-device at the XR system, and a second set of augmentation workloads are performed by one or more devices remote from the XR system, such as cloud devices, smart home devices, edge device, personal computing devices, or any other suitable remote computing device.

**[00100]** When the selected augmentation workload is determined to be a remote augmentation workload, process 1000 can progress to block 1018. When the selected augmentation workload is determined to be an on-device augmentation workload, process 1000 can progress to block 1012.

**[00101]** At block 1012, process 1000 can analyze next stage data using one or more second stage models. Example next stage data can include one or more of the captured environment data, a processed version of the captured environment data (e.g., after processing by the one or more first stage models), or any combination thereof. For example, next stage data can include a portion of a captured image that contains a detected and classified object, a high-level object label classified by one or more first stage model(s), and any other suitable data.

**[00102]** In some implementations, next stage data can comprise predefined data for the selected augmentation workload. For example, where the augmentation workload comprises a multi-stage workload with the first stage predicting a high-level object category (e.g., animal, book, landmark, art work, object type, etc.) and the second stage predicting a granular object category (e.g., dog breed, specific book, such as title, author, identification number, etc., location and title of landmark, specific painting, such as title and painter, etc.), the next stage data can include portion(s) of one or more captured frames of the object, the classified high-level category, and any other suitable data.

**[00103]** At block 1014, process 1000 can generate augmentation data using the second stage model(s). For example, the second stage model(s) can generate, using the next stage data, one or more of: a granular classification for the detected object (e.g., dog breed for an object classified as a dog, book title and author for an object classified as a book, landmark/POI name and location for an object classified as a landmark/monument, etc.), a visual augment, an audio augment, or any other suitable augmentation data.

**[00104]** In some implementations, the second stage model(s) can comprise machine learning models trained/configured to complete the selected augmentation workload. For example, one of the second stage models can be trained/configured to predict a dog breed using image(s) of a dog. In another example, one of the second stage models can be trained/configured to predict specific book information (e.g., identification number, such as ISBN, title, author, etc.). In another example, one of the second stage models can be trained/configured to predict specific POI information (e.g., name and location of landmark/monument, etc.).

**[00105]** In some implementations, second stage model(s) can be configured to augment visual data, such as video frames. For example, the first stage model(s) may categorize an object that triggers a rule for generating a visual augment (e.g., an object with a QR code or other suitable code, a brand of clothing, furniture, or other suitable object(s), etc.), and the selected augmentation workload may generate such an augment.

In this example, second stage model(s) can comprise one or more machine learning models trained/configured to generate an augmented version of captured visual data (e.g., captured frames), such as Generative Adversarial Networks (GANs), encoder/decoder models, and the like.

**[00106]** At block 1016, process 1000 can output the augmentation data to a user via the XR system. For example, the augmentation data output to the user can include a visual display (e.g., overlay) that includes granular object classification and additional information (e.g., dog breed classification and relevant information about the dog breed, book title, author, and ISBN number, name and location of a POI, along with relevant factual data, etc.). In some implementations, the augmentation data can be an augmented version of captured visual data (e.g., captured frames with a visual augmentation), and the output of the augmentation data can include displaying the augmented version of the visual frame(s) to the user.

**[00107]** At block 1018, process 1000 can transmit next stage data to a remote computing device. Example next stage data can include one or more of the captured environment data, a processed version of the captured environment data (e.g., after processing by the one or more first stage models), or any combination thereof. For example, next stage data can include a portion of a captured image that contains a detected and classified object, a high-level object label classified by one or more first stage model(s), and any other suitable data.

**[00108]** For example, the remote computing device can be a cloud computing device, an edge computing device, a smart home computing device, a personal computing device (e.g., personal computer, smartphone, etc.), or any other suitable computing device remote from the XR system. The next stage data can be communicated over any suitable network link (e.g., WIFI, Bluetooth, near-field communication, cellular network, the Internet, etc.).

**[00109]** At block 1020, process 1002 can receive the next stage data from the XR system. For example, the remote computing device can receive the next stage data. At block 1022, process 1002 can analyze the next stage data using one or more second stage model(s). In some implementations, next stage data can comprise predefined data for the selected augmentation workload. For example, where the augmentation workload comprises a multi-stage workload with the first stage predicting a high-level object category (e.g., animal, book, landmark, object type, etc.) and the second stage predicting a granular object category (e.g., dog breed, specific book, such as title, author, identification number, etc., location and title of landmark, etc.), the next stage data can include portion(s) of one or more captured frames of the object, the classified high-level category, and any other suitable data. Second stage model(s) can analyze the next stage data as part of the selected augmentation workflow.

**[00110]** At block 1024, process 1002 can generate augmentation data using the second stage model(s). For example, the second stage model(s) can generate, using the next stage data, one or more of: a granular classification for the detected object (e.g., dog breed for an object classified as a dog, book title and author for an object classified as a book, landmark/POI name and location for an object classified as a landmark/monument, etc.), a visual augment, an audio augment, or any other suitable augmentation data.

**[00111]** In some implementations, the second stage model(s) can comprise machine learning models trained/configured to complete the selected augmentation workload. For example, one of the second stage models can be trained/configured to predict a dog breed using image(s) of a dog. In another example, one of the second stage models can be trained/configured to predict specific book information (e.g., identification number, such as ISBN, title, author, etc.). In another example, one of the second stage models can be trained/configured to predict specific POI information (e.g., name and location of landmark/monument, etc.).

**[00112]** In some implementations, second stage model(s) can be configured to augment visual data, such as video frames. For example, the first stage model(s) may categorize an object that triggers a rule for generating a visual augment (e.g., an object with a QR code or other suitable code, a brand of clothing, furniture, or other suitable object(s), etc.), and the selected augmentation workload may generate such an augment.

**[00113]** At block 1026, process 1002 can transmit the augmentation data to the XR system. For example, the augmentation data generated by the second stage model(s) can be transmitted from the remote computing device to the XR system. At block 1028, process 1000 can receive the augmentation data from the remote computing device. For example, the XR system can receive the augmentation data.

**[00114]** At block 1030, process 1000 can output the augmentation data to a user via the XR system. For example, the augmentation data output to the user via the XR system can include a visual display (e.g., overlay) that includes granular object classification and additional information (e.g., dog breed classification and relevant information about the dog breed, book title, author, and ISBN number, name and location of a POI, along with relevant factual data, etc.). In some implementations, the augmentation data can be an augmented version of captured visual data (e.g., captured frames with a visual augmentation), and the output of the augmentation data can include displaying the augmented version of the visual frame(s) to the user.

**[00115]** Figure 11 is a block diagram illustrating an overview of devices on which some implementations of the disclosed technology can operate. The devices can comprise hardware components of a device 1100 as shown and described herein. Device 1100 can include one or more input devices 1120 that provide input to the Processor(s) 1110 (e.g., CPU(s), GPU(s), HPU(s), etc.), notifying it of actions. The actions can be mediated by a hardware controller that interprets the signals received from the input device and communicates the information to the processors 1110 using a communication protocol. Input devices 1120 include, for example, a mouse, a keyboard, a touchscreen, an infrared sensor, a touchpad, a wearable input device, a camera- or image-based input device, a microphone, or other user input devices.

**[00116]** Processors 1110 can be a single processing unit or multiple processing units in a device or distributed across multiple devices. Processors 1110 can be coupled to other hardware devices, for example, with the use of a bus, such as a PCI bus or SCSI bus. The processors 1110 can communicate with a hardware controller for devices, such as for a display 1130. Display 1130 can be used to display text and graphics. In some implementations, display 1130 provides graphical and textual visual feedback to a user. In some implementations, display 1130 includes the input device as part of the display, such as when the input device is a touchscreen or is equipped with an eye direction monitoring system. In some implementations, the display is separate from the input device. Examples of display devices are: an LCD display screen, an LED display screen, a projected, holographic, or augmented reality display (such as a heads-up display device or a head-mounted device), and so on. Other I/O devices 1140 can also be coupled to the processor, such as a network card, video card, audio card, USB, firewire or other external device, camera, printer, speakers, CD-ROM drive, DVD drive, disk drive, or Blu-Ray device.

**[00117]** In some implementations, the device 1100 also includes a communication device capable of communicating wirelessly or wire-based with a network node. The communication device can communicate with another device or a server through a network using, for example, TCP/IP protocols. Device 1100 can utilize the communication device to distribute operations across multiple network devices.

**[00118]** The processors 1110 can have access to a memory 1150 in a device or distributed across multiple devices. A memory includes one or more of various hardware devices for volatile and non-volatile storage, and can include both read-only and writable memory. For example, a memory can comprise random access memory (RAM), various caches, CPU registers, read-only memory (ROM), and writable non-volatile memory, such as flash memory, hard drives, floppy disks, CDs, DVDs, magnetic storage devices, tape drives, and so forth. A memory is not a propagating signal divorced from underlying hardware; a memory is thus non-transitory. Memory 1150 can include program memory 1160 that stores programs and software, such as an operating system 1162, XR Interactive Model System 1164, and other application programs 1166. Memory 1150 can also include data memory 1170, which can be provided to the program memory 1160 or any element of the device 1100.

**[00119]** Some implementations can be operational with numerous other computing system environments or configurations. Examples of computing systems, environments, and/or configurations that may be suitable for use with the technology include, but are not limited to, personal computers, server computers, handheld or laptop devices, cellular telephones, wearable electronics, gaming consoles, tablet devices, multiprocessor systems, microprocessor-based systems, set-top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, or the like.

**[00120]** Figure 12 is a block diagram illustrating an overview of an environment 1200 in which some implementations of the disclosed technology can operate. Environment 1200 can include one or more client computing devices 1205A-D, examples of which can include device 1100. Client computing devices 1205 can operate in a networked environment using logical connections through network 1230 to one or more remote computers, such as a server computing device.

**[00121]** In some implementations, server 1210 can be an edge server which receives client requests and coordinates fulfillment of those requests through other servers, such as servers 1220A-C. Server computing devices 1210 and 1220 can comprise computing systems, such as device 1100. Though each server computing device 1210 and 1220 is displayed logically as a single server, server computing devices can each be a distributed computing environment encompassing multiple computing devices located at the same or at geographically disparate physical locations. In some implementations, each server 1220 corresponds to a group of servers.

**[00122]** Client computing devices 1205 and server computing devices 1210 and 1220 can each act as a server or client to other server/client devices. Server 1210 can connect to a database 1215. Servers 1220A-C can each connect to a corresponding database 1225A-C. As discussed above, each server 1220 can correspond to a group of servers, and each of these servers can share a database or can have their own database. Databases 1215 and 1225 can warehouse (e.g., store) information. Though databases 1215 and 1225 are displayed logically as single units, databases 1215 and 1225 can each be a distributed computing environment encompassing multiple computing devices, can be located within their corresponding server, or can be located at the same or at geographically disparate physical locations.

**[00123]** Network 1230 can be a local area network (LAN) or a wide area network (WAN), but can also be other wired or wireless networks. Network 1230 may be the Internet or some other public or private network. Client computing devices 1205 can be connected to network 1230 through a network interface, such as by wired or wireless communication. While the connections between server 1210 and servers 1220 are shown as separate connections, these connections can be any kind of local, wide area, wired, or wireless network, including network 1230 or a separate public or private network.

**[00124]** Embodiments of the disclosed technology may include or be implemented in conjunction with an artificial reality system. Artificial reality or extra reality (XR) is a form of reality that has been adjusted in some manner before presentation to a user, which may include, e.g., a virtual reality (VR), an augmented reality (AR), a mixed reality (MR), a hybrid reality, or some combination and/or derivatives thereof. Artificial reality content may include completely generated content or generated content combined with captured content (e.g., real-world photographs). The artificial reality content may include video, audio, haptic feedback, or some combination thereof, any of which may be presented in a single channel or in multiple channels (such as stereo video that produces a three-dimensional effect to the viewer). Additionally, in some embodiments, artificial reality may be associated with applications, products, accessories, services, or some combination thereof, that are, e.g., used to create content in an artificial reality and/or used in (e.g., perform activities in) an artificial reality. The artificial reality system that provides the artificial reality content may be implemented on various platforms, including a head-mounted display (HMD) connected to a host computer system, a standalone HMD, a mobile device or computing system, a "cave" environment or other projection system, or any other hardware platform capable of providing artificial reality content to one or more viewers.

**[00125]** "Virtual reality" or "VR," as used herein, refers to an immersive experience where a user's visual input is controlled by a computing system. "Augmented reality" or

"AR" refers to systems where a user views images of the real world after they have passed through a computing system. For example, a tablet with a camera on the back can capture images of the real world and then display the images on the screen on the opposite side of the tablet from the camera. The tablet can process and adjust or "augment" the images as they pass through the system, such as by adding virtual objects. "Mixed reality" or "MR" refers to systems where light entering a user's eye is partially generated by a computing system and partially composes light reflected off objects in the real world. For example, a MR headset could be shaped as a pair of glasses with a pass-through display, which allows light from the real world to pass through a waveguide that simultaneously emits light from a projector in the MR headset, allowing the MR headset to present virtual objects intermixed with the real objects the user can see. "Artificial reality," "extra reality," or "XR," as used herein, refers to any of VR, AR, MR, or any combination or hybrid thereof. Additional details on XR systems with which the disclosed technology can be used are provided in U.S. Patent Application No. 17/170,839, titled "INTEGRATING ARTIFICIAL REALITY AND OTHER COMPUTING DEVICES," filed 2/8/2021 and now issued as U.S. Patent No. 11,402,964 on 8/2/2022, which is herein incorporated by reference.

**[00126]** Those skilled in the art will appreciate that the components and blocks illustrated above may be altered in a variety of ways. For example, the order of the logic may be rearranged, substeps may be performed in parallel, illustrated logic may be omitted, other logic may be included, etc. As used herein, the word "or" refers to any possible permutation of a set of items. For example, the phrase "A, B, or C" refers to at least one of A, B, C, or any combination thereof, such as any of: A; B; C; A and B; A and C; B and C; A, B, and C; or multiple of any item such as A and A; B, B, and C; A, A, B, C, and C; etc. Any patents, patent applications, and other references noted above are incorporated herein by reference. Aspects can be modified, if necessary, to employ the systems, functions, and concepts of the various references described above to provide yet further implementations. If statements or subject matter in a document incorporated by reference conflicts with statements or subject matter of this application, then this application shall control.

## CLAIMS

I/We claim:

1. A method for disambiguating a selection through zoom and gaze on an artificial reality device, the method comprising:

rendering a zoomed-out view of content having a plurality of interactive mechanisms on the artificial reality device;

detecting a selection gesture by a user of the artificial reality device;

in response to detecting the selection gesture, rendering a zoomed-in view of the content, the zoomed-in view including at least one interactive mechanism of the plurality of interactive mechanisms;

capturing a gaze direction of the user of the artificial reality device, the gaze direction being toward an interactive mechanism of the at least one interactive mechanism; and

in response to the gaze direction being toward the interactive mechanism, selecting the interactive mechanism.

2. A method for delivering a shortcut associated with a real-world object in an artificial reality environment, the method comprising:

identifying a real-world object in a real-world environment of a user by performing object recognition on one or more images of the real-world environment, the real-world object having a corresponding object anchor registered on an artificial reality device, the artificial reality device being configured to display at least one of an augmented reality experience, a mixed reality experience, or both;

identifying the shortcut associated with the real-world object using the object anchor; and

performing at least one of an action on the artificial reality device, an action in the real-world environment, or both, based on the identified shortcut.

3. A method for artificial reality (XR) augmentation using a distributed model architecture, the method comprising:

capturing, at a XR system, environment data of a real-world environment, wherein the environment data comprises at least visual frames;

detecting, using one or more first stage models, objects in the environment data;

selecting, using the one or more first stage models, an augmentation workload for the environment data, wherein the augmentation workload is selected from among a predefined set of augmentation workloads;

providing next stage environment data, wherein:

the next stage environment data comprises one or more of the captured environment data, a processed version of the captured environment data after processing by the one or more first stage models, or any combination thereof, and

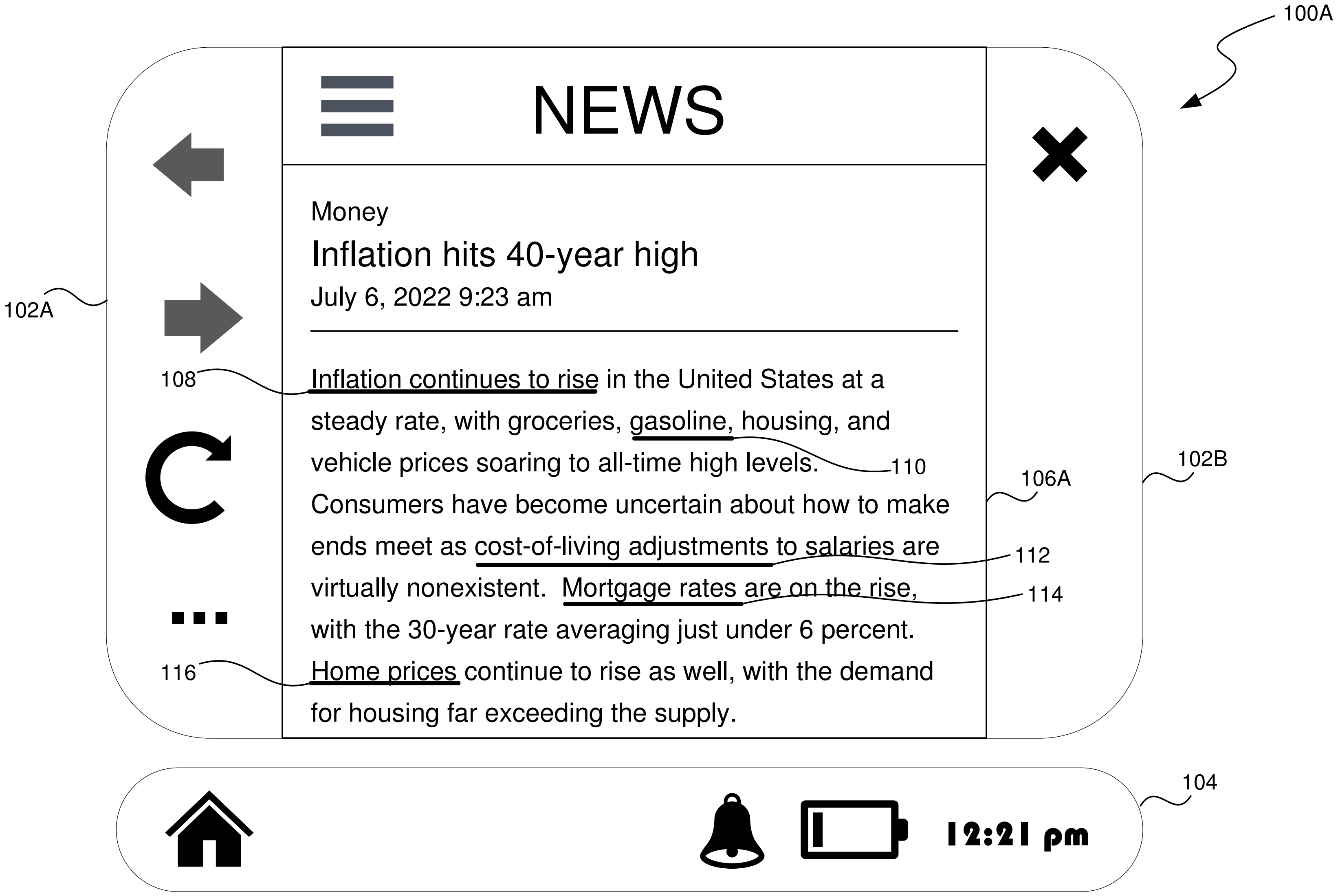
the providing causes performing, using one or more second stage models, the selected augmentation workload on the next stage environment data;

receiving augmentation data generated using the one or more second stage models and the next stage environment data; and

outputting, to a user of the client XR system, a XR environment and an environment augment using the received augmentation data.

## ABSTRACT

In some implementations, the technology can render a dense layout of interactive mechanisms (e.g., selectable text and/or graphics) that are responsive to low accuracy input methods on the XR device. In some implementations, an XR device can associate a shortcut with the physical object (e.g., an action relative to the physical object, an option to perform an action relative to the physical object, etc.). In some implementations, a workload manager can select an augmentation workload using one or more of captured environment data (e.g., captured visual frames, audio, etc.), output from the initial stage model(s), any other suitable data.



NEWS

Money

Inflation hits 40-year high

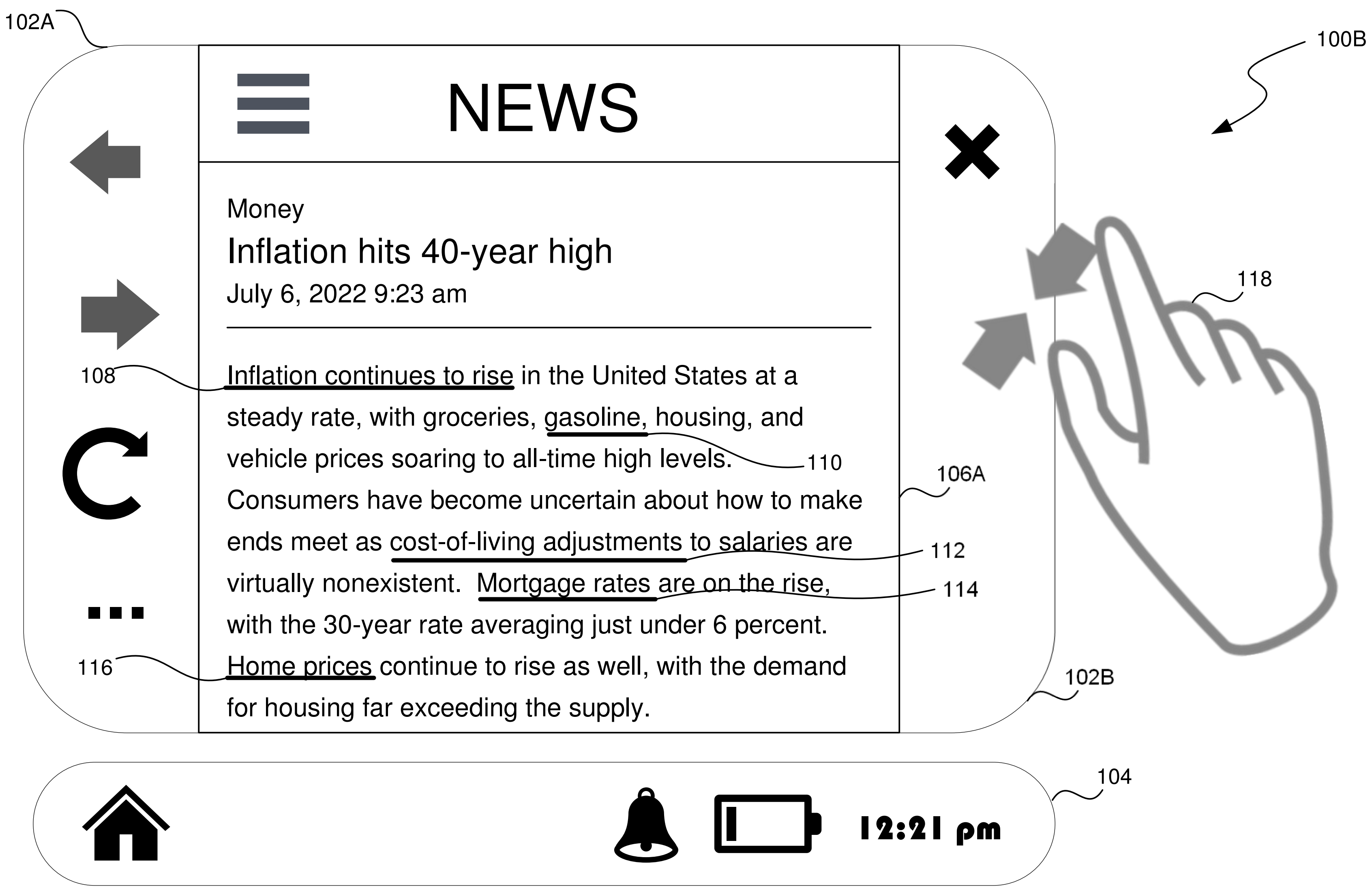
July 6, 2022 9:23 am

Inflation continues to rise in the United States at a steady rate, with groceries, gasoline, housing, and vehicle prices soaring to all-time high levels. Consumers have become uncertain about how to make ends meet as cost-of-living adjustments to salaries are virtually nonexistent. Mortgage rates are on the rise, with the 30-year rate averaging just under 6 percent. Home prices continue to rise as well, with the demand for housing far exceeding the supply.

*FIG. 1A*

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01



NEWS

Money

Inflation hits 40-year high

July 6, 2022 9:23 am

Inflation continues to rise in the United States at a steady rate, with groceries, gasoline, housing, and vehicle prices soaring to all-time high levels. Consumers have become uncertain about how to make ends meet as cost-of-living adjustments to salaries are virtually nonexistent. Mortgage rates are on the rise, with the 30-year rate averaging just under 6 percent. Home prices continue to rise as well, with the demand for housing far exceeding the supply.

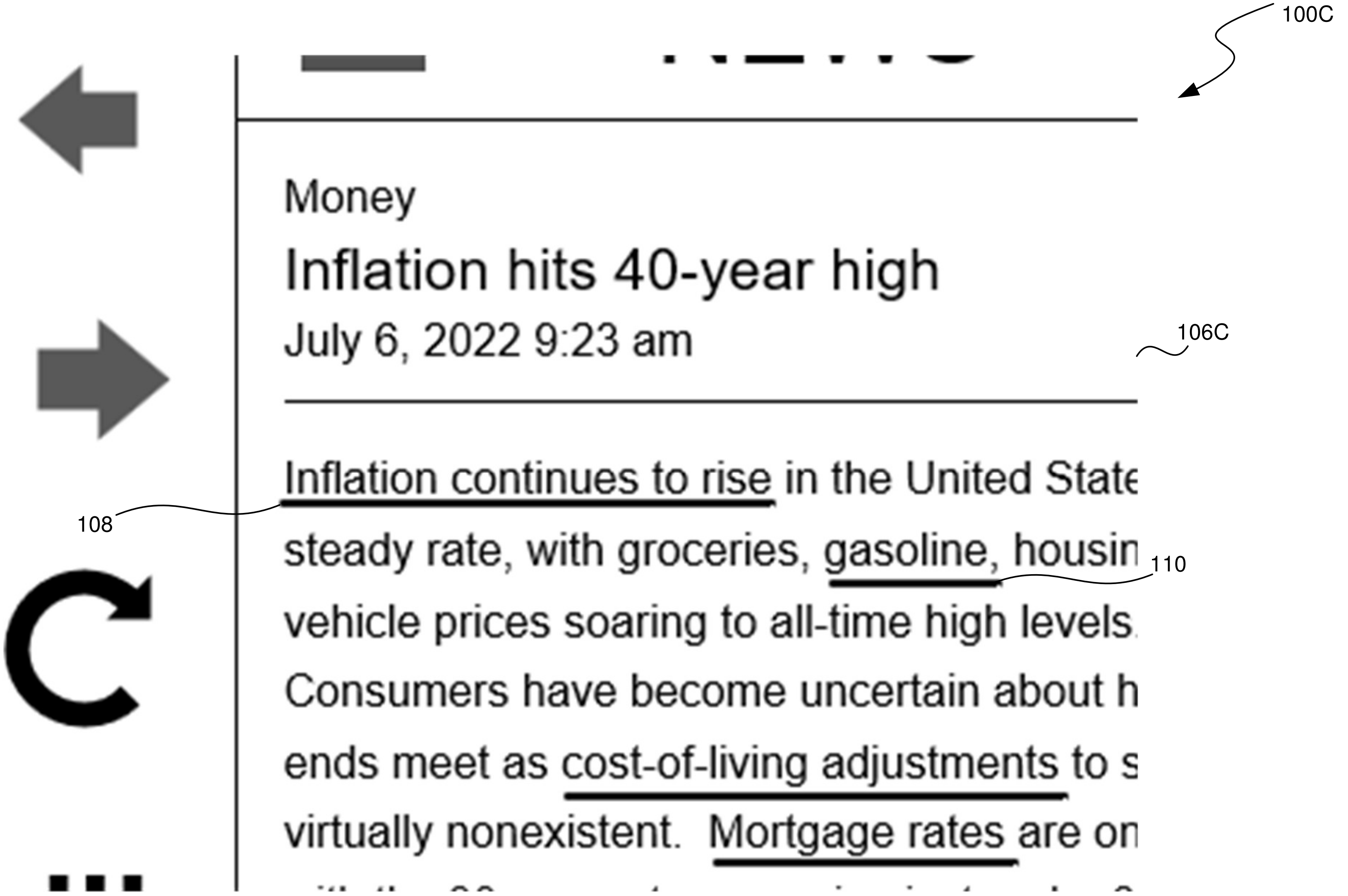
12:21 pm

*FIG. 1B*

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models



Money

Inflation hits 40-year high

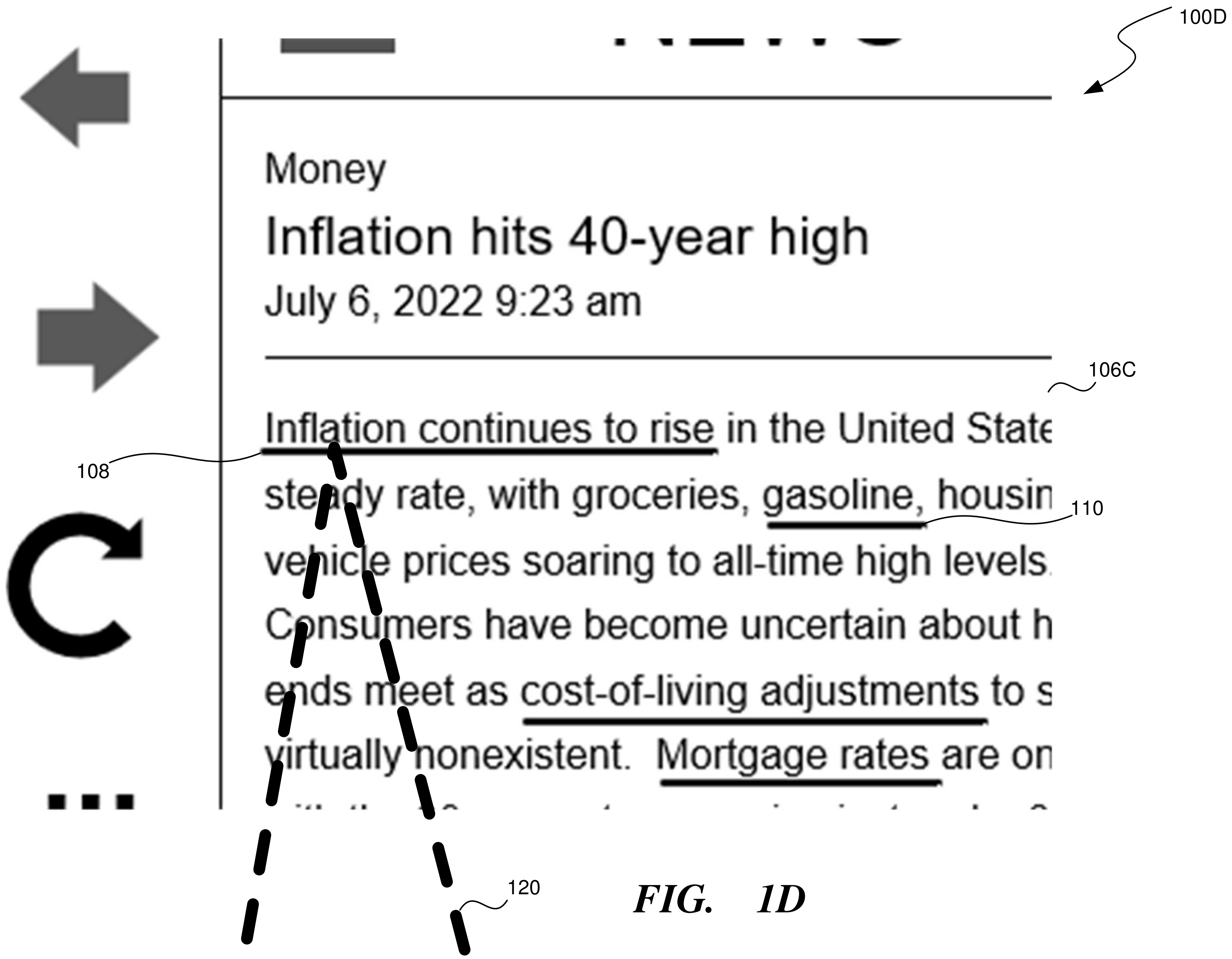
July 6, 2022 9:23 am

Inflation continues to rise in the United State
steady rate, with groceries, gasoline, housin
vehicle prices soaring to all-time high levels.
Consumers have become uncertain about h
ends meet as cost-of-living adjustments to s
virtually nonexistent. Mortgage rates are on

FIG. 1C

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models



Money

Inflation hits 40-year high

July 6, 2022 9:23 am

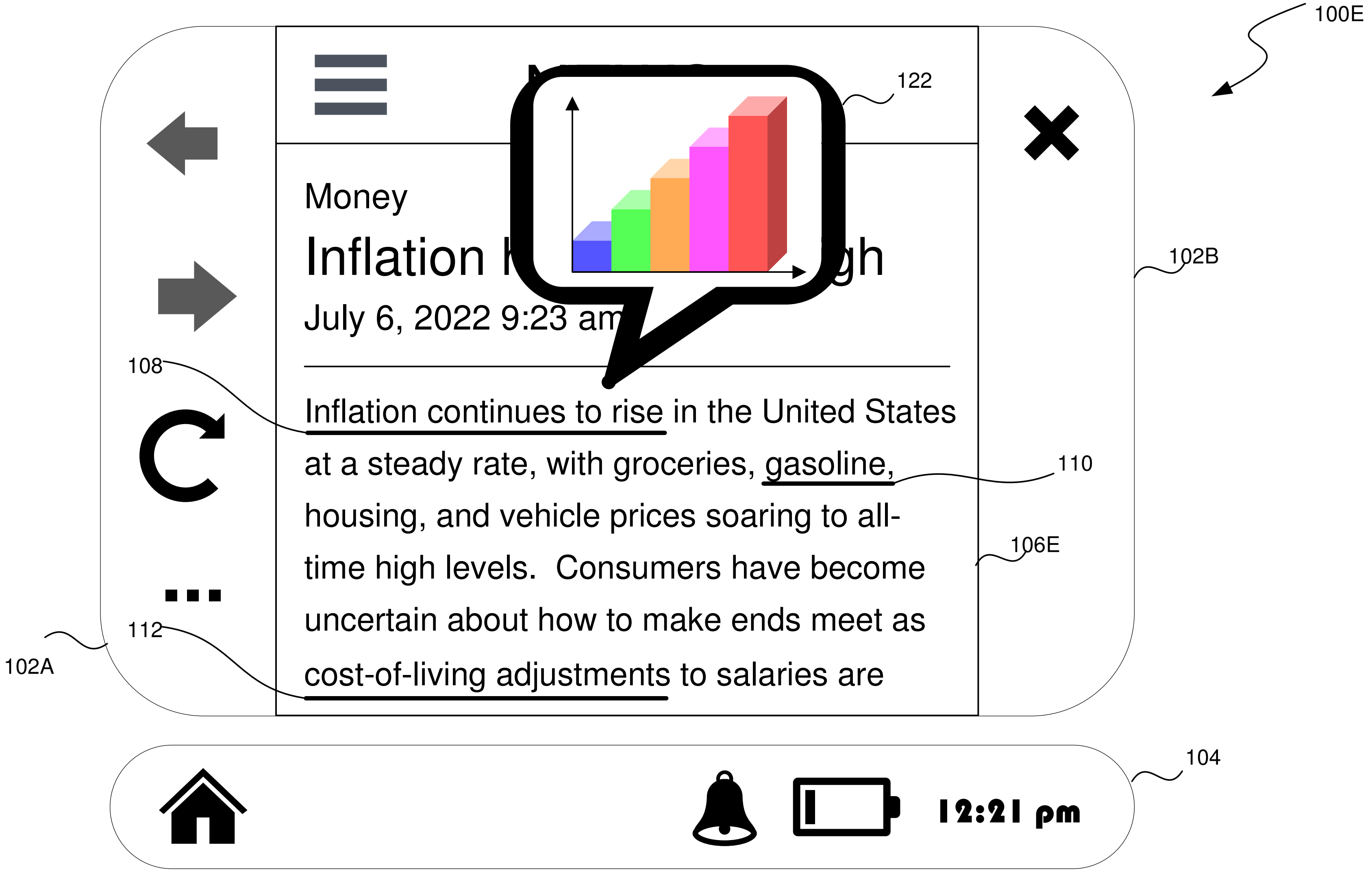
Inflation continues to rise in the United State
steady rate, with groceries, gasoline, housin
vehicle prices soaring to all-time high levels.
Consumers have become uncertain about h
ends meet as cost-of-living adjustments to s
virtually nonexistent.  Mortgage rates are on

*FIG. 1D*

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01



Money

Inflation h gh

July 6, 2022 9:23 am

Inflation continues to rise in the United States at a steady rate, with groceries, gasoline, housing, and vehicle prices soaring to all-time high levels. Consumers have become uncertain about how to make ends meet as cost-of-living adjustments to salaries are

FIG. 1E

**First Inventor:** Pol PLA I CONESA **Attorney Docket No.:** 3589-0167DP01
**Title:** XR Interactive Models

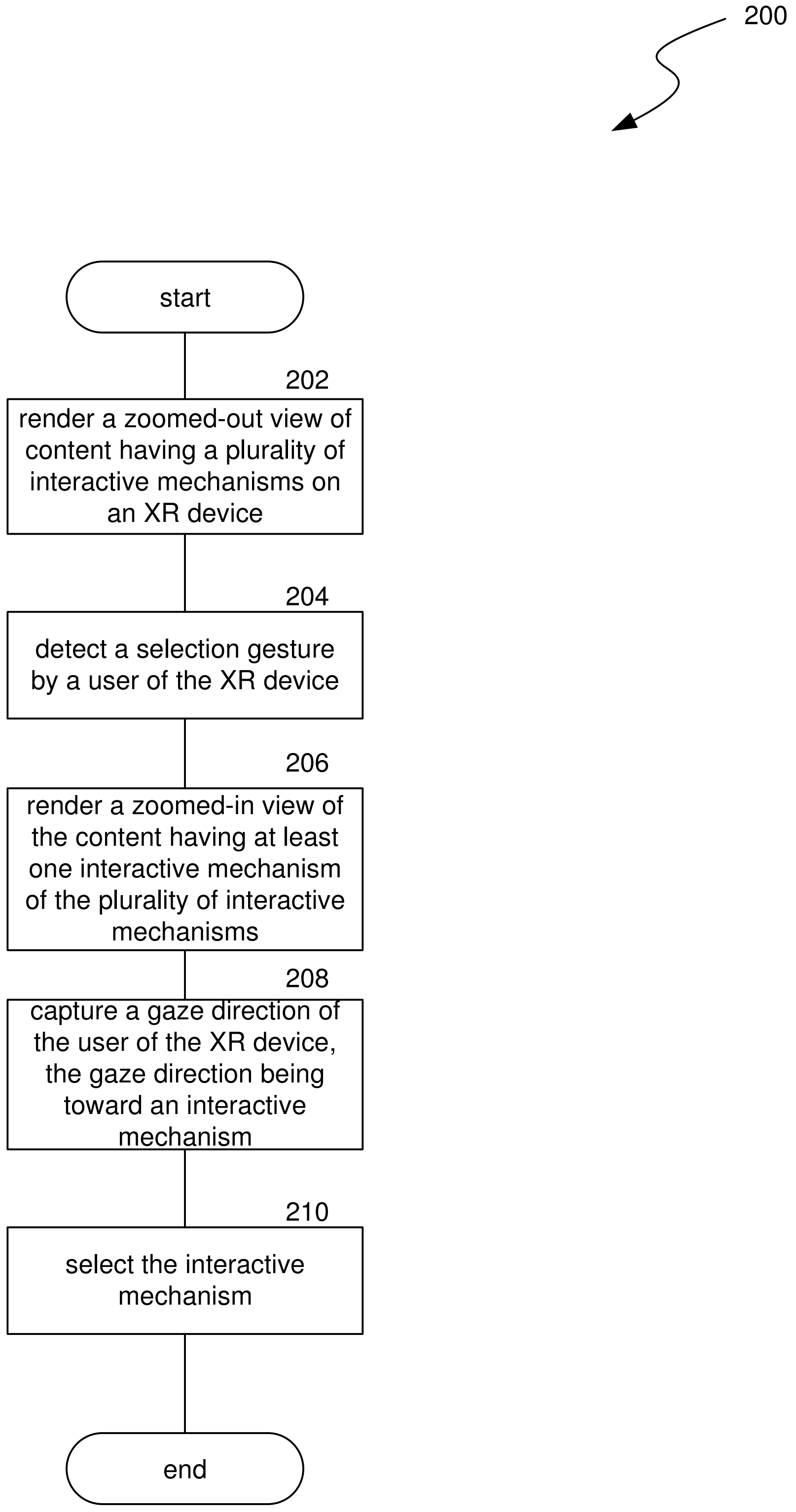

200
start
202
render a zoomed-out view of content having a plurality of interactive mechanisms on an XR device
204
detect a selection gesture by a user of the XR device
206
render a zoomed-in view of the content having at least one interactive mechanism of the plurality of interactive mechanisms
208
capture a gaze direction of the user of the XR device, the gaze direction being toward an interactive mechanism
210
select the interactive mechanism
end

*FIG. 2*

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01

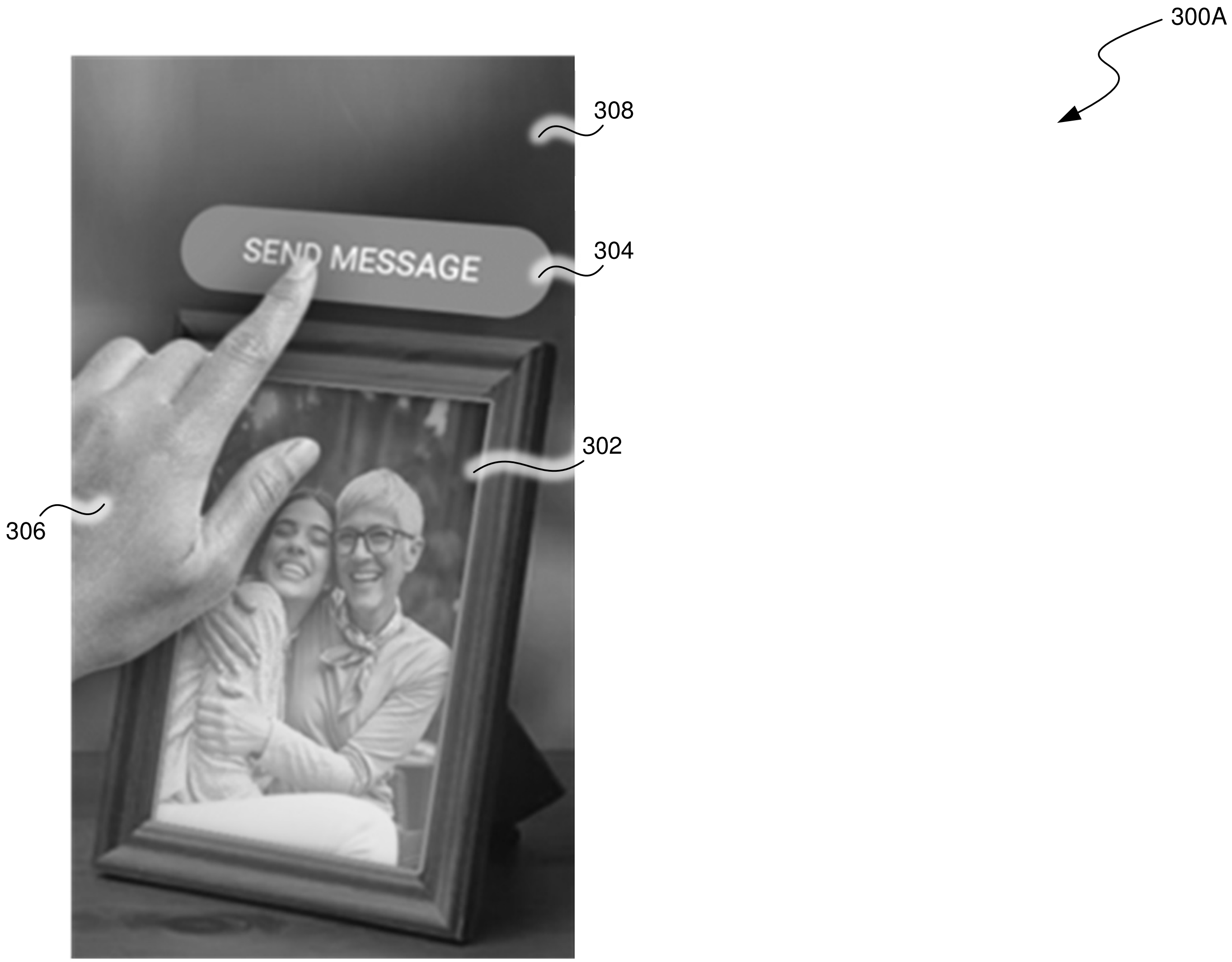
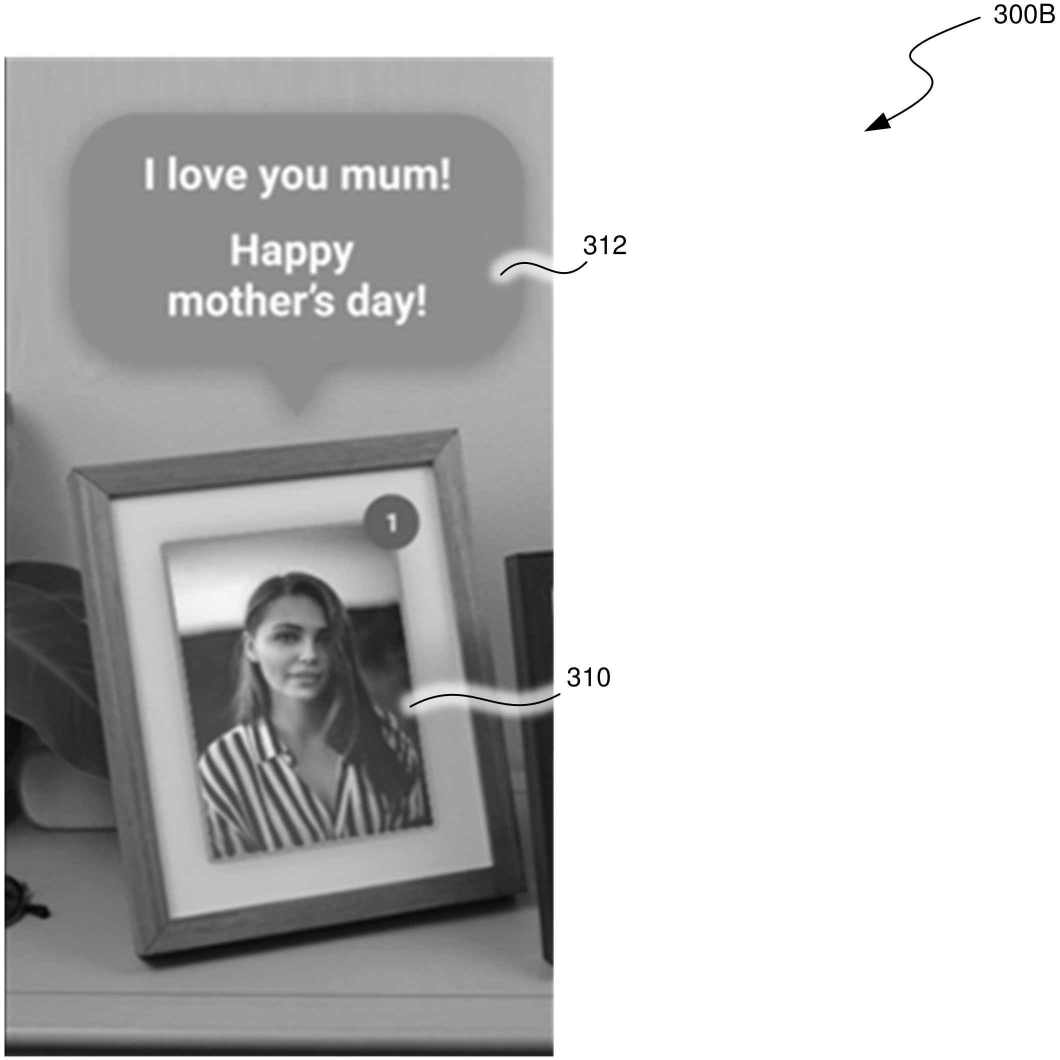


FIG. 3A

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models



I love you mum!

Happy mother's day!

*FIG. 3B*

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01

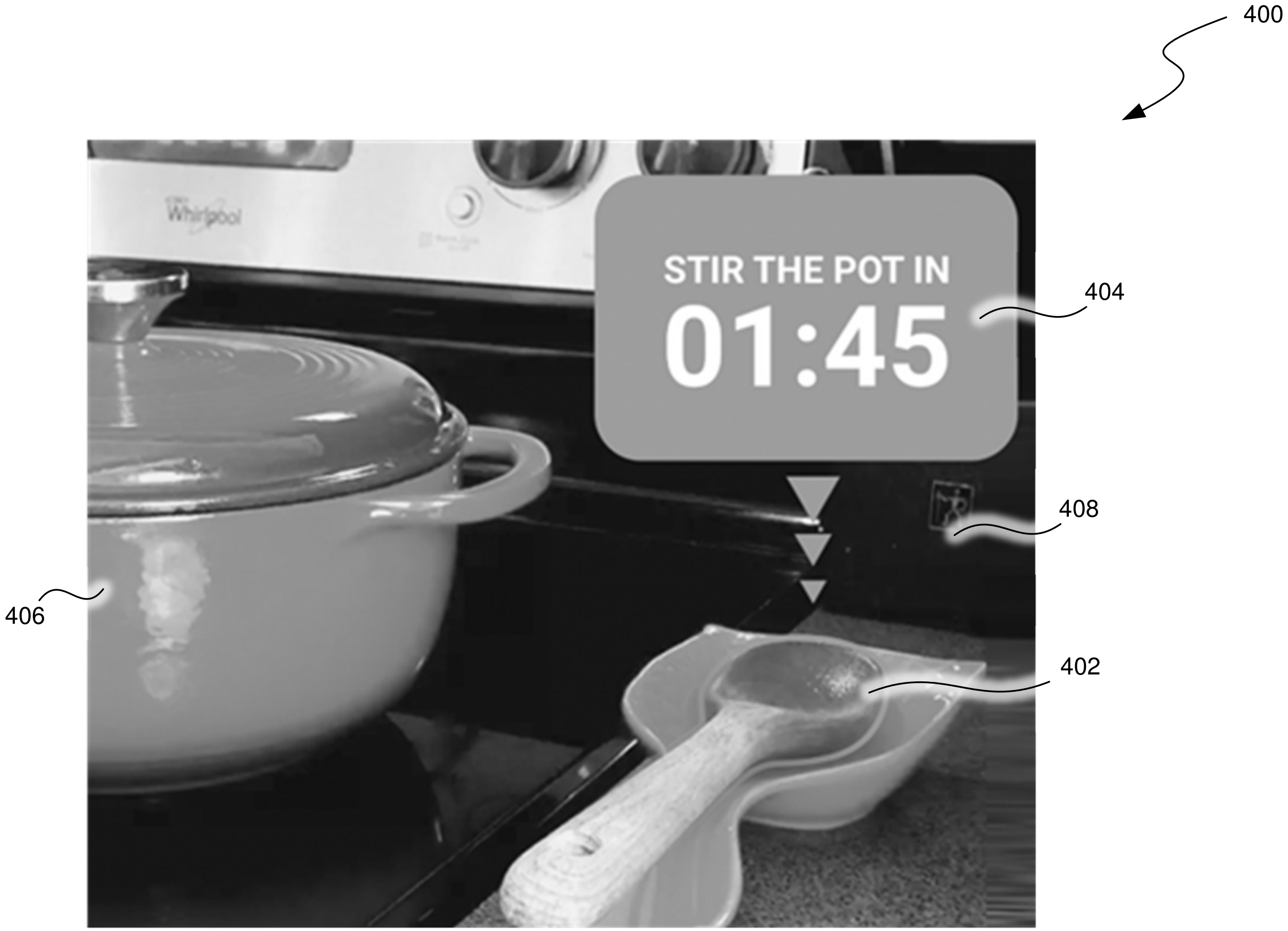


*FIG. 4*

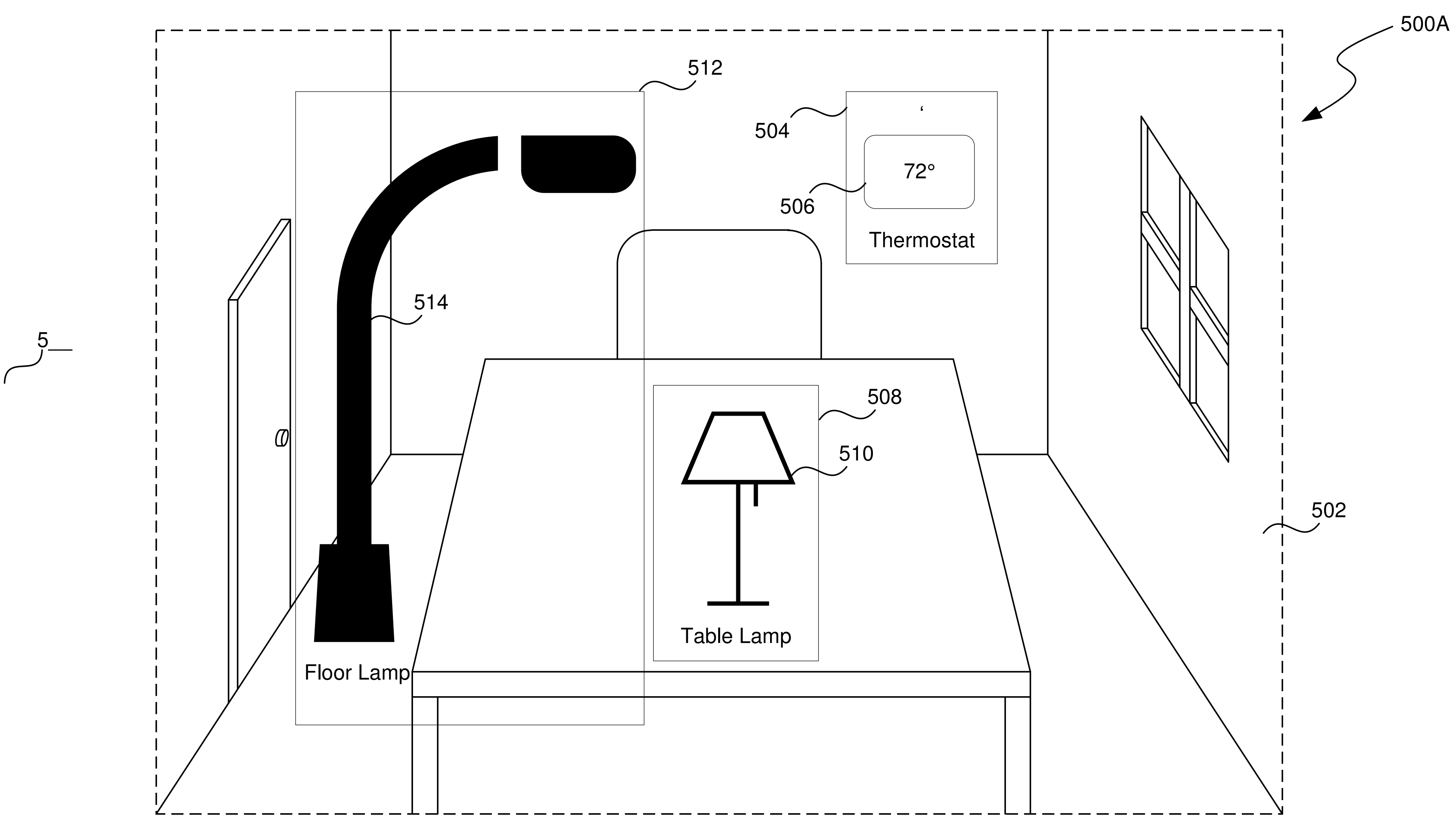


FIG. 5A

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

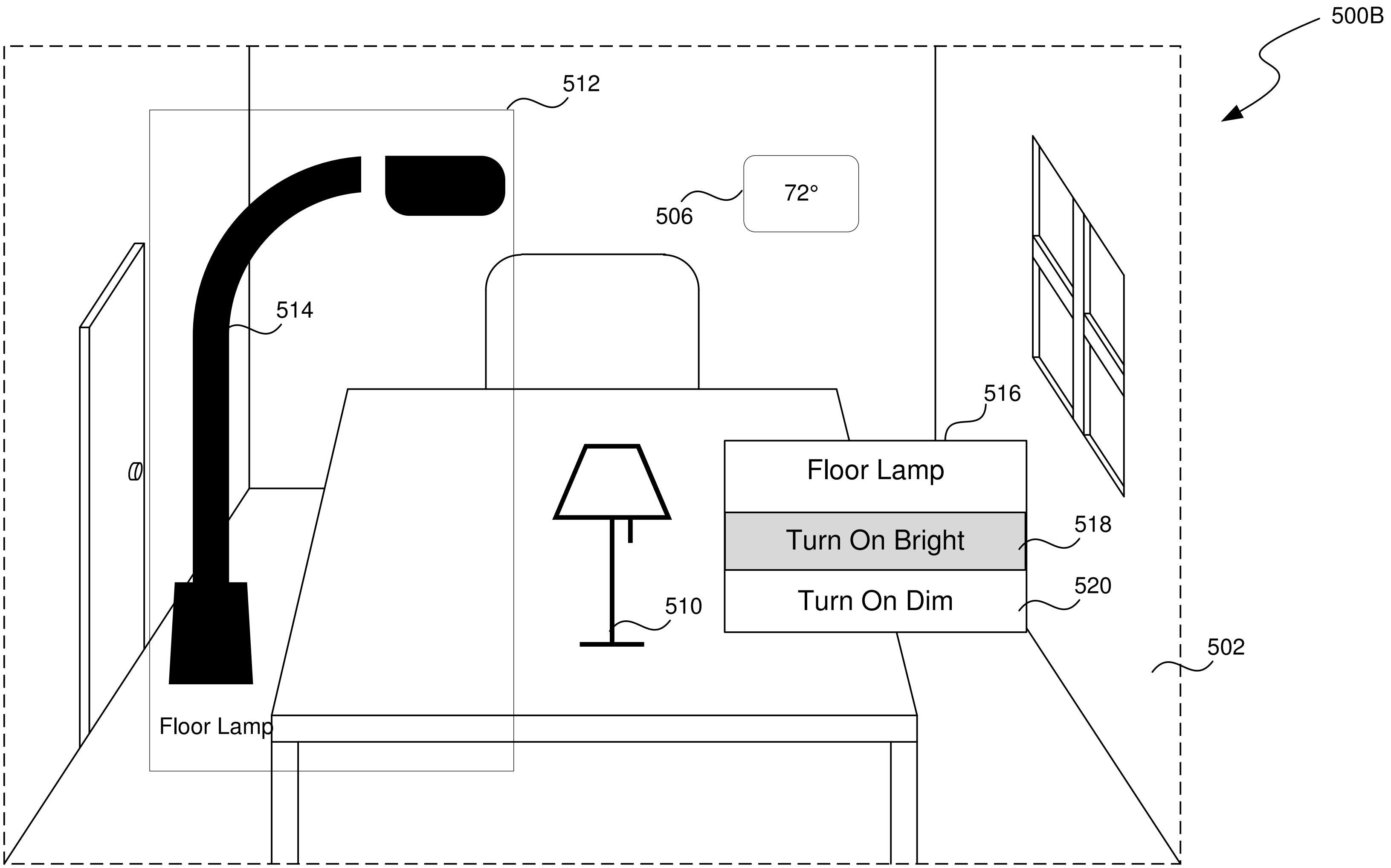


FIG. 5B

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

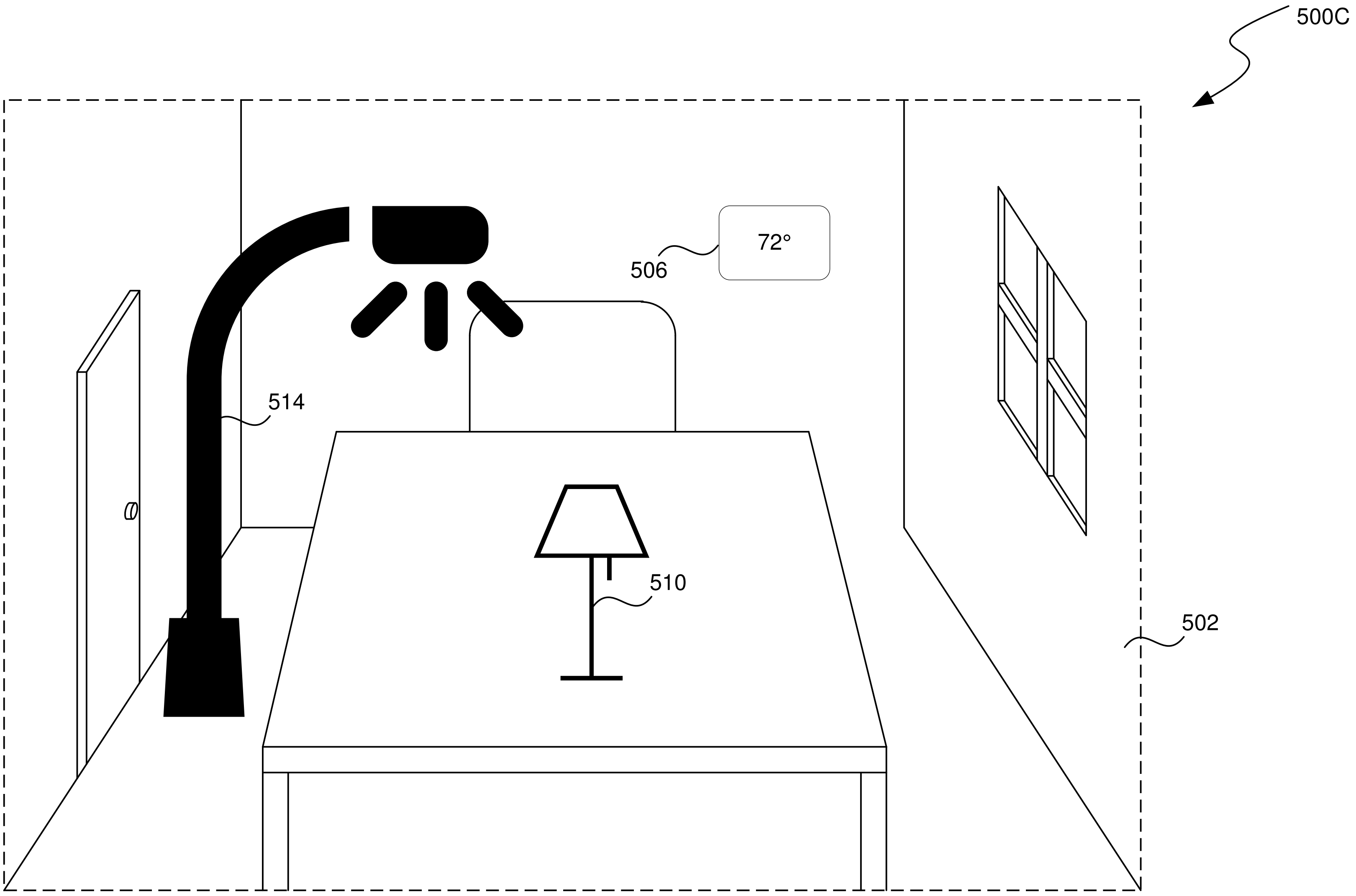


*FIG. 5C*

**Attorney Docket No.:** 3589-0167DP01

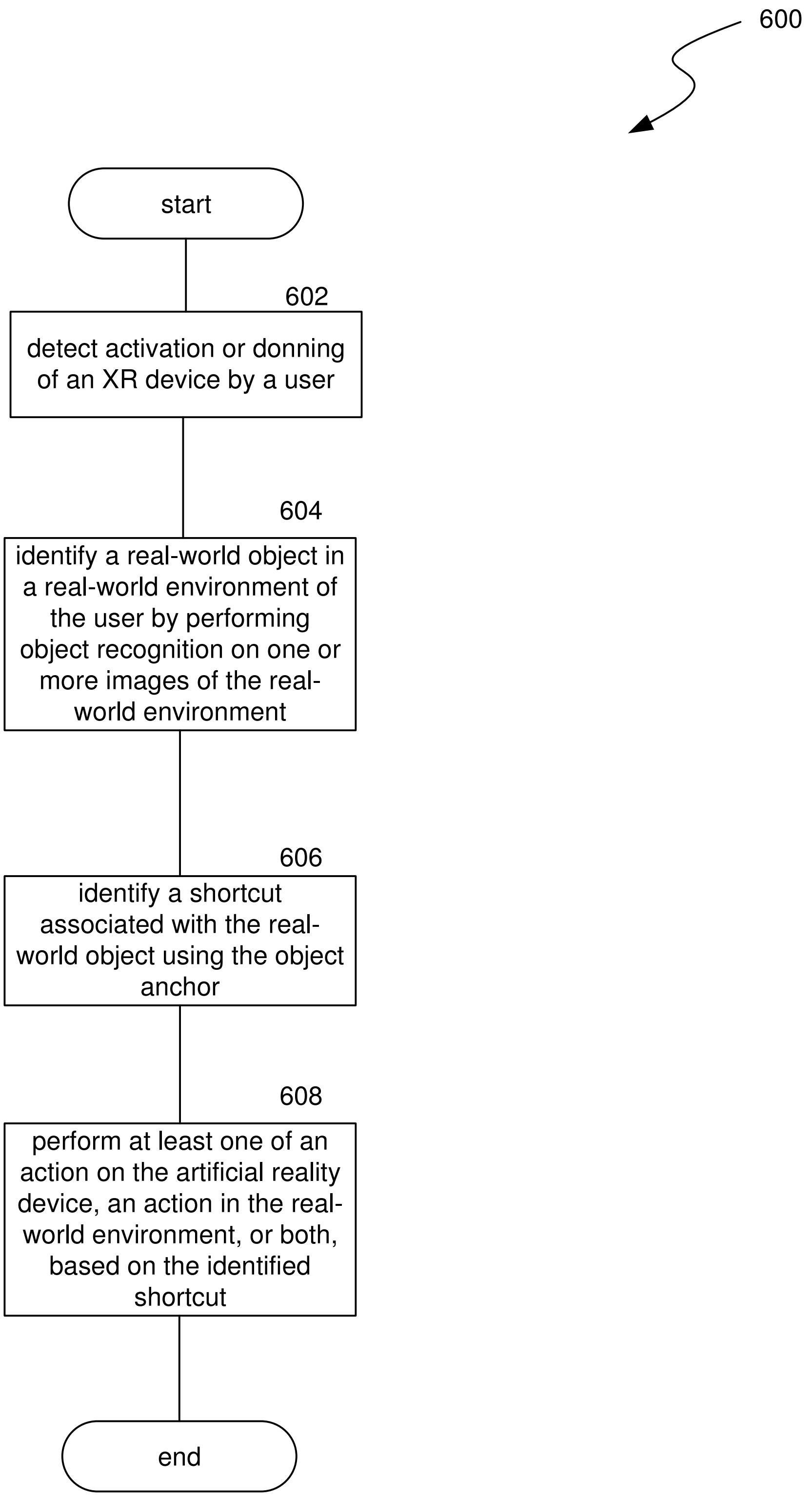

600
start
602
detect activation or donning of an XR device by a user
604
identify a real-world object in a real-world environment of the user by performing object recognition on one or more images of the real-world environment
606
identify a shortcut associated with the real-world object using the object anchor
608
perform at least one of an action on the artificial reality device, an action in the real-world environment, or both, based on the identified shortcut
end

*FIG. 6*

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

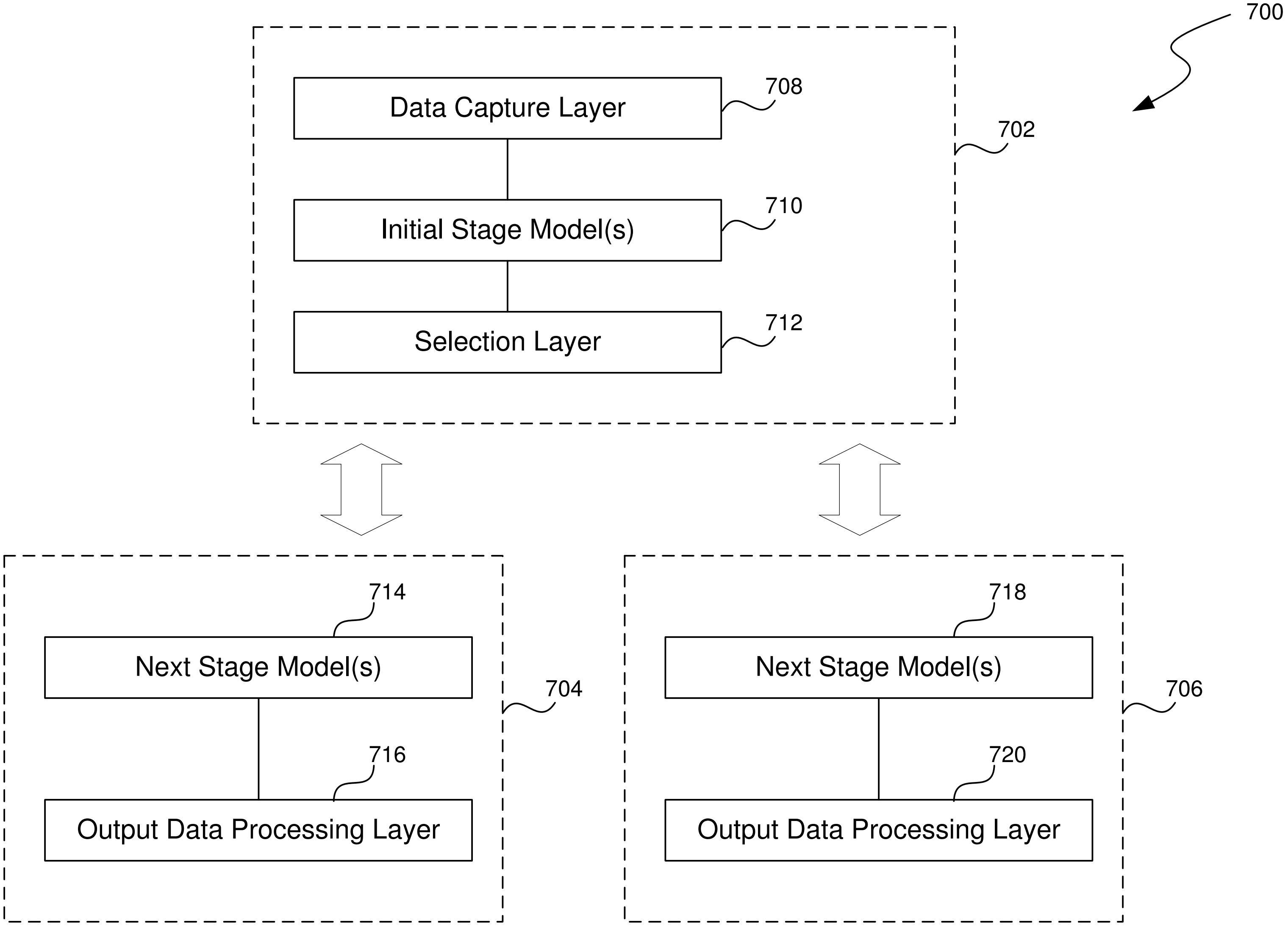
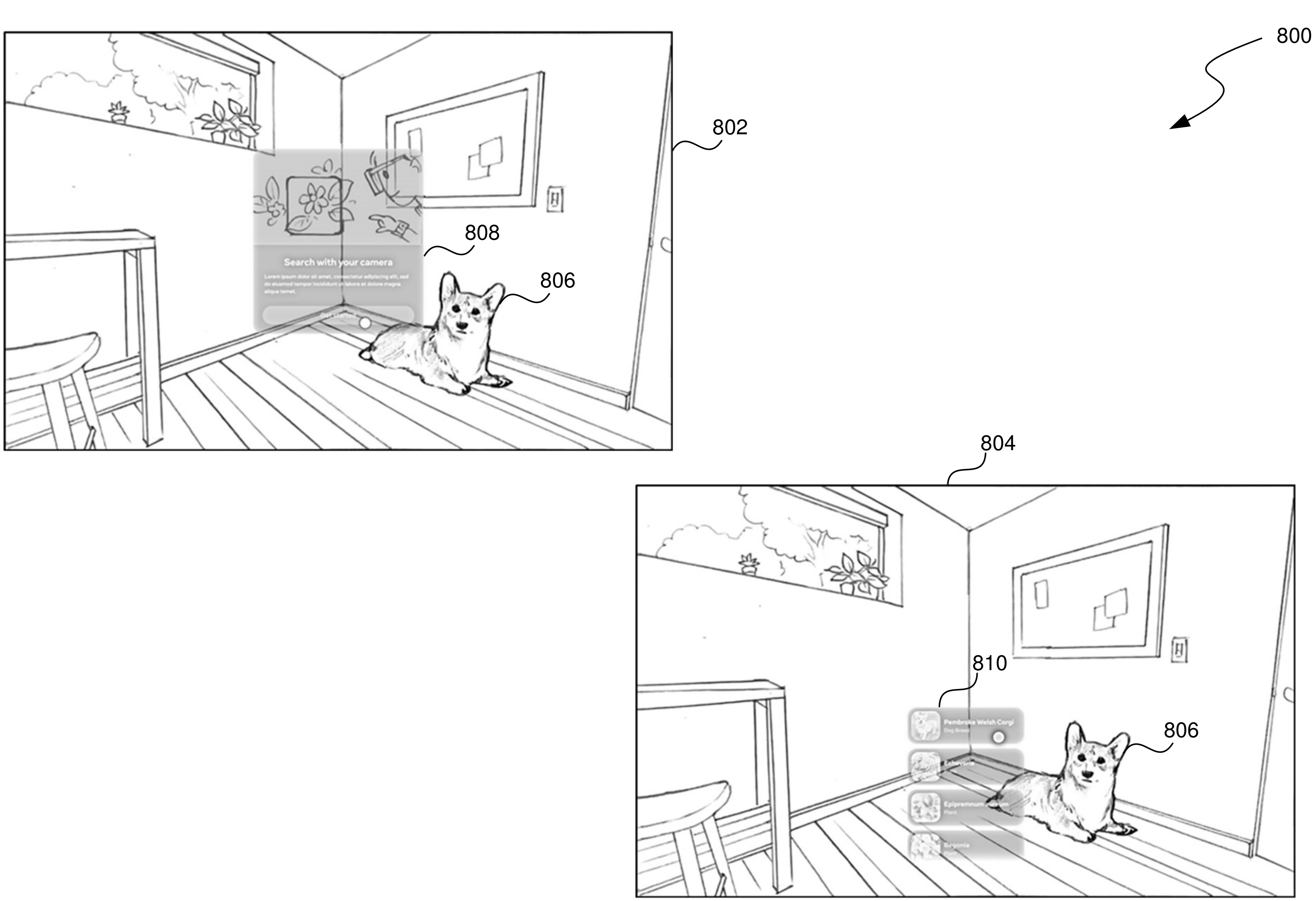


FIG. 7

**Attorney Docket No.:** 3589-0167DP01



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua tamet.

***FIG. 8***

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

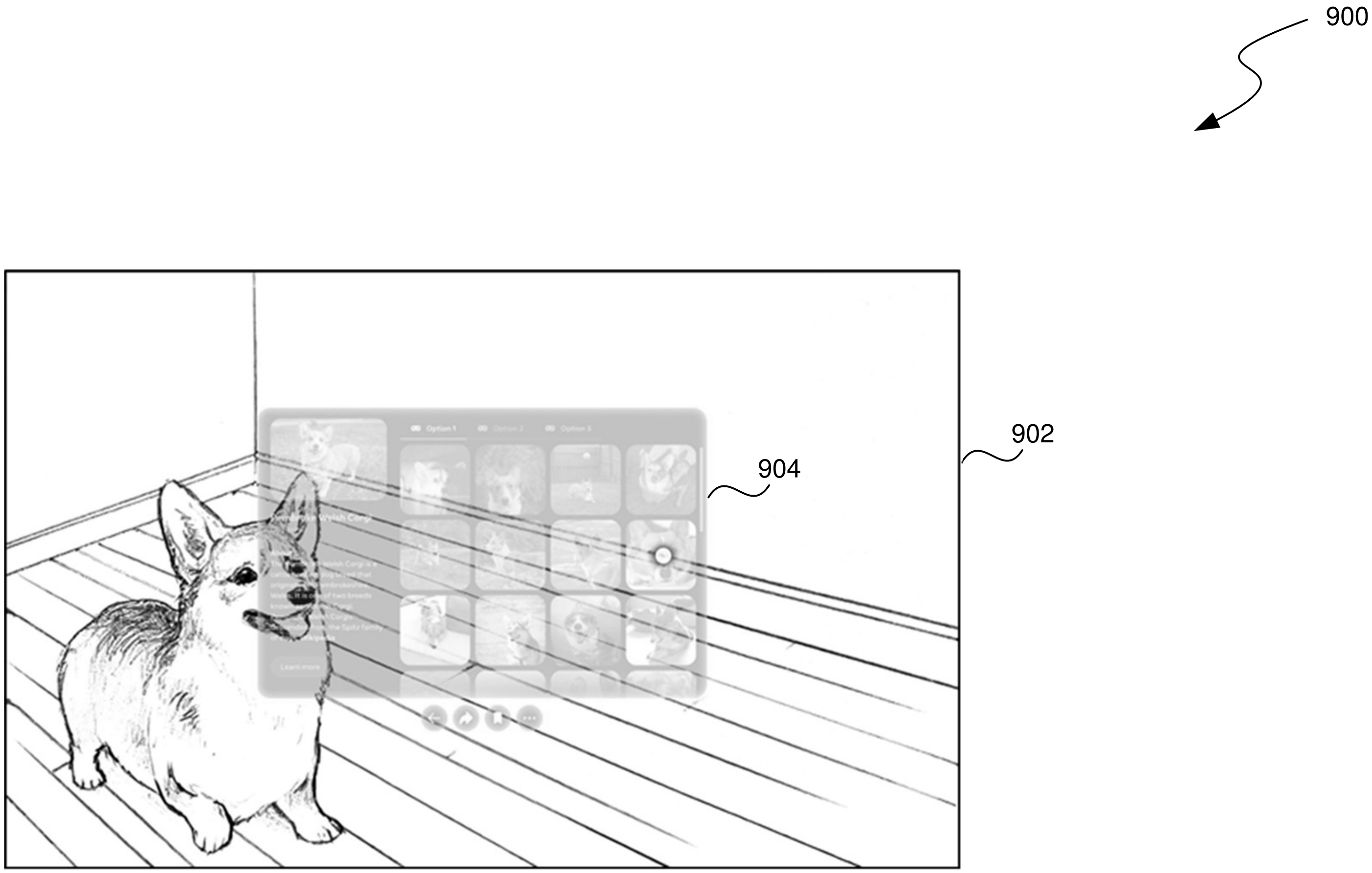


*FIG. 9*

**Attorney Docket No.:** 3589-0167DP01

**First Inventor:** Pol PLA I CONESA **Attorney Docket No.:** 3589-0167DP01
**Title:** XR Interactive Models

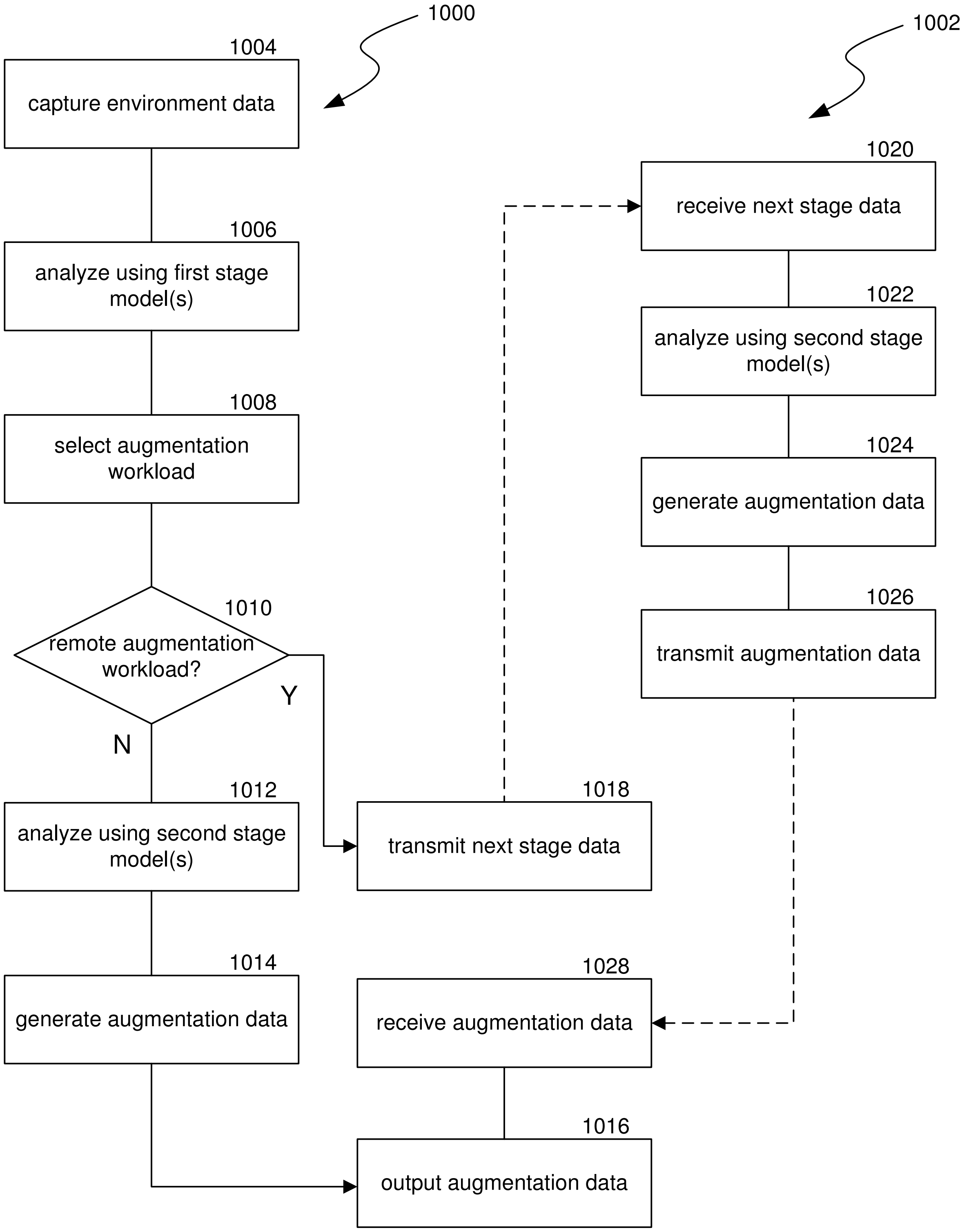

1000
1002
1004
capture environment data
1006
analyze using first stage model(s)
1008
select augmentation workload
1010
remote augmentation workload?
Y
N
1012
analyze using second stage model(s)
1014
generate augmentation data
1018
transmit next stage data
1020
receive next stage data
1022
analyze using second stage model(s)
1024
generate augmentation data
1026
transmit augmentation data
1028
receive augmentation data
1016
output augmentation data

***FIG. 10***

**First Inventor:** Pol PLA I CONESA **Attorney Docket No.:** 3589-0167DP01
**Title:** XR Interactive Models

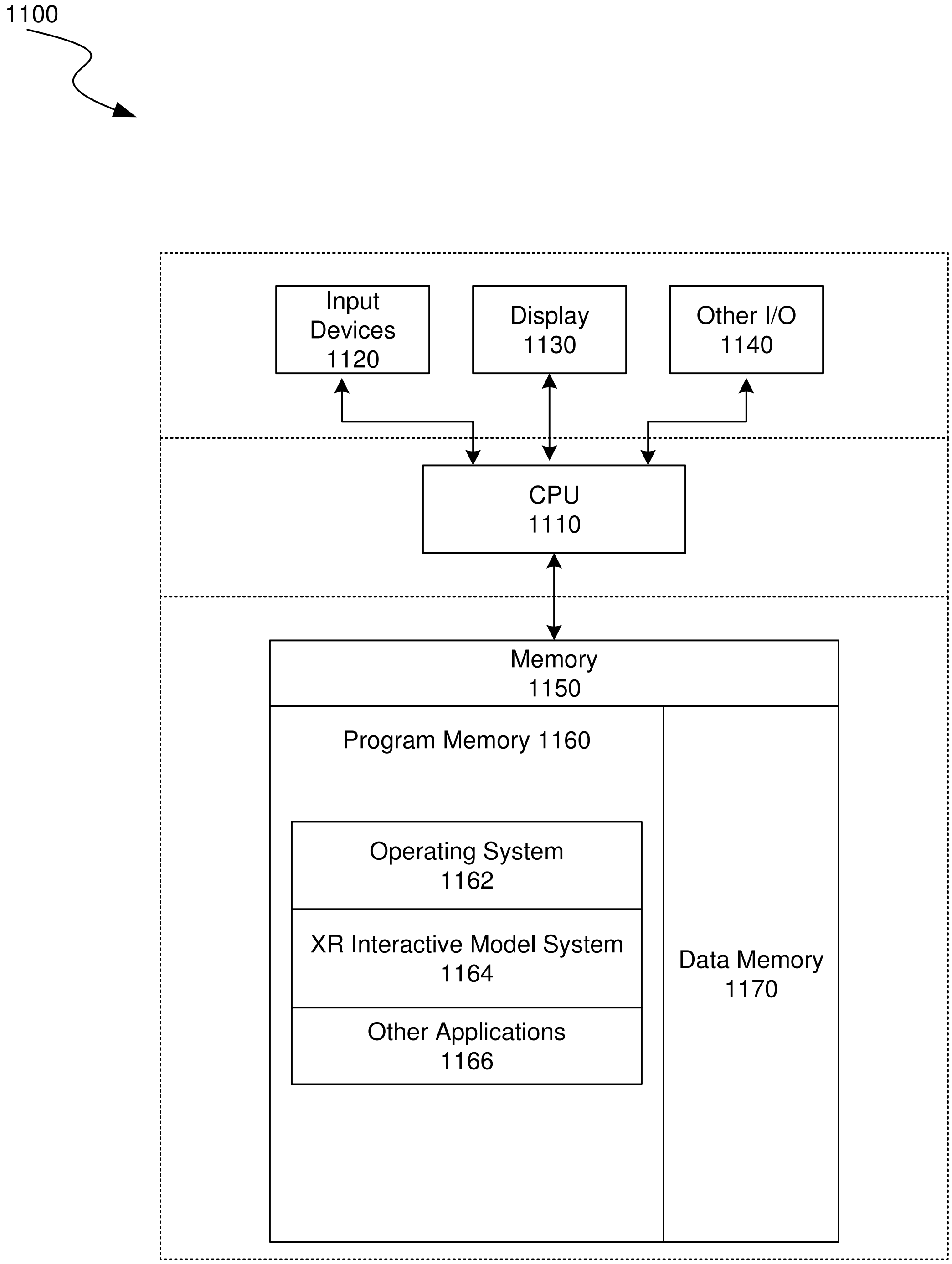


*FIG. 11*

**First Inventor:** Pol PLA I CONESA
**Title:** XR Interactive Models

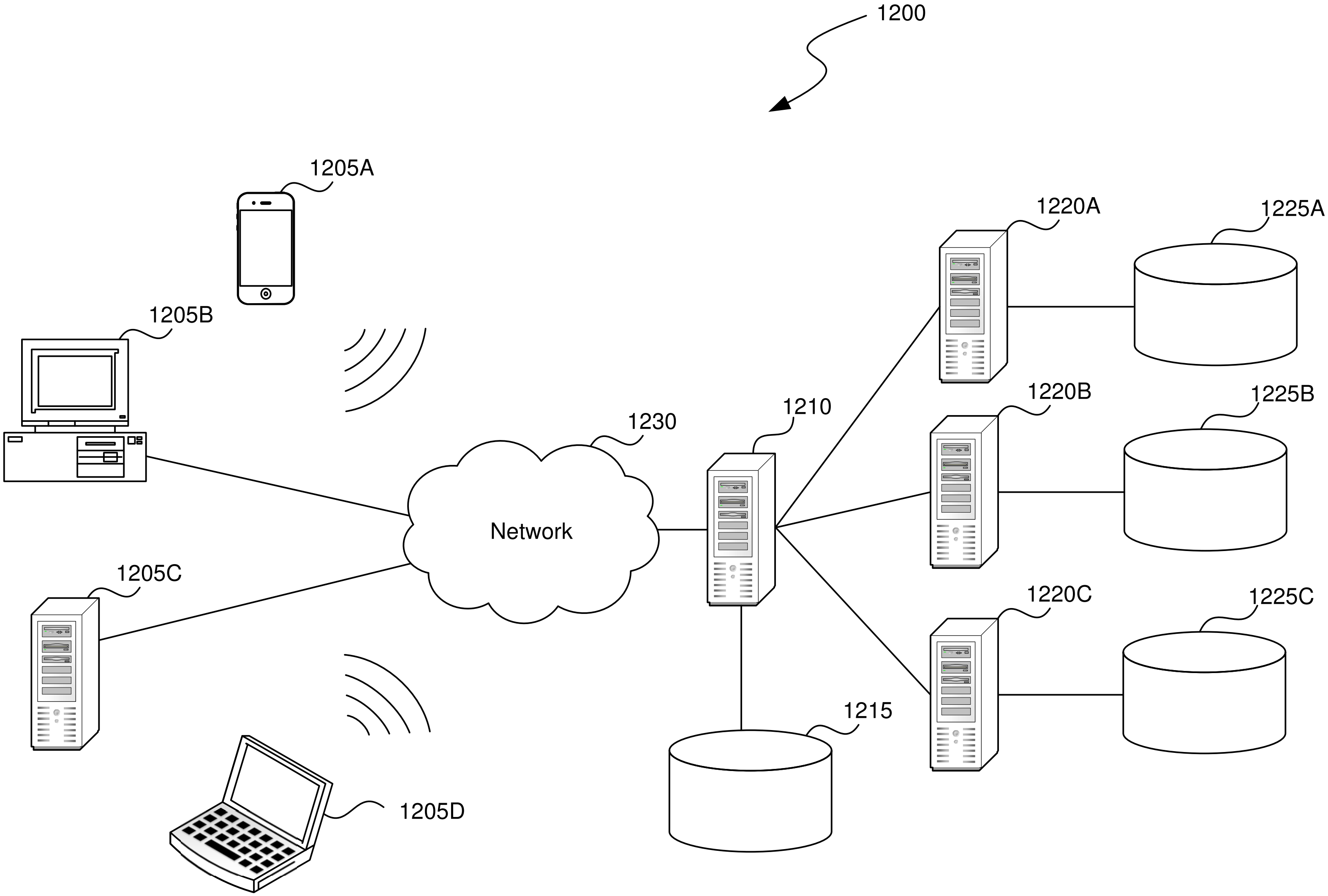


**FIG. 12**

**Attorney Docket No.:** 3589-0167DP01