will be appreciated that the terms 'non-luminous object' and 'passive non-luminous object' are used interchangeably.

[0091] At a first step **S901**, a plurality of images of a non-luminous object being held by a user is obtained. The images may be obtained in any of the previously described manners. For example, the images may be received at e.g. a communication interface of a video game playing device or intermediate device, from an external camera via a wired or wireless connection. Alternatively, it may be that the camera is integral to, or attached to, the video game playing device (e.g. attached to an HMD) and so the images are received from such a camera. More generally, step **S901** may comprise receiving the images at an input unit of a computing device, as described previously in relation to FIG. **7**. The video images correspond to video images captured of the user in real or near-real time (which may have delays associated with propagation and processing), for controlling the operation of a video game playing device.

[0092] At a second step **S902**, pixels corresponding to the non-luminous object are detected in the obtained images. As before, this detection is based on the content of the image corresponding to the object being held and not based on a physical identifier (such as a QR code) that has been added to the object. As described previously, step **S902** may involve detecting a plurality of non-luminous objects being held by the same or different users. In some examples, detecting pixels in the obtained images as corresponding to the non-luminous object comprises detecting a contour in the obtained images as corresponding to the non-luminous object being held by the user. In such examples, step **S902** may comprise filtering one or more colours from the obtained images known not to correspond to the object being held by the user, prior to performing the contour detection. A binary mask may then be generated for each filtered image, and contour detection performed for each image. In such examples, the contour (or contours) detected as having a maximal area relative to the other contours in the obtained images may be identified as corresponding to the object being held by the user.

[0093] In some examples, step **S902** may comprise detecting a plurality of contours in the obtained images and identifying a largest of the detected contours as corresponding to the non-luminous object being held by the use (as described previously in relation to FIGS. **2** to **6**). As described previously, in some examples, a single user may be holding two objects (e.g. of the same type) in each hand. In such examples, step **S902** may involve detecting at least two non-luminous objects being held by the user, with each object being detected based on one or more contours detected as corresponding to that object. A location within the image representation of each object may be identified based on the one or more contours detected for that object. If, for example, a single contour is detected for an object, then the location may correspond to a central point or location. If, for example, two contours are detected for a given object, the location may be determined as location located between the detected contours or e.g. in a larger of the two contours.

[0094] In alternative or additional examples, step **S902** may comprise inputting the obtained images to a machine learning model that has been trained to identify 'every day' inanimate items in images.

[0095] At a third step **S903**, changes in pose of the non-luminous object are detected based on the image content of the obtained images. The change in pose of the passive non-luminous object may be based on at least one of a (i) contour detection operation and (ii) the output of a machine learning model that has been trained to detect the poses of objects in 2D images of such objects.

[0096] As described previously, step **S903** may involve detecting changes in at least one of the (i) position, (ii) orientation and (iii) area of the contour (relative to corresponding default positions, orientations and areas). In examples where a single user is detected as holding two objects (e.g. one in each hand), the poses of the object may be determined as described previously. For example, step **S903** may involve detecting a central point or region for each object, and detecting changes in position and orientation of the two objects based on how the position (or length) and orientation of a line connecting the two centre points changes (see FIGS. **5** and **6**).

[0097] In examples where machine learning is used, it may be that detecting the pose (step **S903**) comprises performing 6D pose estimation, as described previously. That is, the obtained images of the user holding the object(s) may be input to a machine learning model that has been trained to perform six-dimensional pose estimation.

[0098] In examples where a user is detected as holing two non-luminous objects, one in each respective hand, step **S903** may involve detecting a change in pose of the at least two non-luminous objects based on changes in orientation of a line intersecting the location detected for each object. The manners in which this may be done were described previously in relation to FIGS. **5** and **6**.

[0099] At a fourth step **S904**, a user input for controlling a virtual object in a video is generated based on the detected changes in pose of the non-luminous object. As described previously, this may involve generating different user inputs depending on whether the rotation, position (x-y) and/or depth (distance from the camera) of the object is detected as changing.

[0100] At a fifth step **S905**, the generated user input is transmitted to a video game processor so as to control the virtual object in the video game in accordance with the generated user input. In some examples, it may be that the user input is generated at a device that is separate from the video game playing device and is thus communicated to the video game playing device via a wired or wireless connection. In other examples, it may be that the user input is generated at the video game playing device itself and is thus communicated to the one or more processors (e.g. CPU) so as to cause the video game to be updated in accordance with the generated user input. Although not shown, it will be appreciated that the method may further comprise updating the display of a virtual environment in accordance with the user input generated in response to the detected changes in pose of the object being held by the user.

[0101] In some examples (not shown) the method may further comprise detecting at least two non-luminous objects being held by different respective users and associating each object to a respective user. This may involve detecting at least two users in the obtained images (using e.g. OpenPose) and determining a relative distance between each user and each object. In this way, it can be determined which object(s) each user is located closest to. Each user may then be associated with the object they are detected as being closest to. In response to having associated each object to a respective user, changes in pose of each object may be tracked