At **1014**, process **1000** can determine whether all visible octants have been account for by determining which points to copy based on which octants represent a one point per pixel resolution. If process **1000** determines that each visible octant has not been accounted for ("NO" at **1014**), process **1000** can move on to a next octant at the lowest resolution which has an octant that has not yet been analyzed. For example, if the analysis started with an octant representing the entire point cloud, process **1000** can move on to a child of that octant (rather than returning to that octant).

Otherwise, if process **1000** determines that each visible octant has been accounted for ("YES" at **1014**), process **1000** can copy points in the selected octants from CPU cache into a frame buffer object of the resolution being analyzed. Note that, because each frame buffer object corresponds to a different resolution, an octant that projects to one point per pixel for a first frame buffer object may project to two points per pixel for a higher resolution frame buffer object.

FIGS. **11**A to **11**C show examples of the same octant at different distances corresponding to different points per pixel in accordance with some aspects of the disclosed subject matter. As shown in FIG. **11**A, the octant projects to about two points per pixel when projected back to the plane represented by the frame buffer object. Accordingly, a lower resolution octant representing the same portion of the scene would be closer to one point per pixel. As shown in FIG. **11**B, when the camera is closer to the octant, the octant projects to about one point per pixel, and the octant is selected for use with that frame buffer object. As shown in FIG. **11**C, when the camera is yet closer to the octant, the octant projects to about one-half point per pixel, and a higher resolution octant representing the same portion of the scene would be closer to one point per pixel. Note that only a single pixel is represented in FIGS. **11**A to **11**C. As can be appreciated from FIGS. **11**A to **11**C, as the camera moves closer to a portion of the scene, higher resolution representations of the scene can be used to render the images presented to user with a lower risk of underdrawing (e.g., leaving gaps) or overdrawing points (e.g., copying excess points that end up being discarded). By contrast, randomly selecting octants can result in overdrawing points, which can in some cases reduce the frame rate that can be achieved due to the presence of more points that are not rendered. Additionally, points in the scene that are relatively close can be represented in more detail by using points from the higher resolution representations of the point cloud (i.e., by selecting higher resolution octants for those portions of the scene), while points in the scene that are relatively far can be represented in less detail (i.e., by selecting lower resolution octants). Note that although only a single octant is shown to avoid overcomplicating the illustration, there can be other visible octants (e.g., closer to or farther from the pixel plane), and the mechanisms described herein can select which resolution of octant(s) to use to represent those spaces.

Note that, although the mechanisms described herein are generally described in connection with rendering a single image, this is merely an example, and in some aspects, two images can be rendered in parallel from two different viewpoints to facilitate presentation of a stereoscopic pair of images. For example, a GPU associated with a head mounted display can render two images in parallel using techniques described herein. In such an example, the visible octants for each image in the stereoscopic pair are generally the same, but in some cases a portion of an octant may be visible in one image, but not in the other.

In some aspects, any suitable computer readable media can be used for storing instructions for performing the functions and/or processes described herein. For example, in some aspects, computer readable media can be transitory or non-transitory. For example, non-transitory computer readable media can include media such as magnetic media (such as hard disks, floppy disks, etc.), optical media (such as compact discs, digital video discs, Blu-ray discs, etc.), semiconductor media (such as RAM, Flash memory, electrically programmable read only memory (EPROM), electrically erasable programmable read only memory (EEPROM), etc.), any suitable media that is not fleeting or devoid of any semblance of permanence during transmission, and/or any suitable tangible media. As another example, transitory computer readable media can include signals on networks, in wires, conductors, optical fibers, circuits, or any suitable media that is fleeting and devoid of any semblance of permanence during transmission, and/or any suitable intangible media.

It should be noted that, as used herein, the term mechanism can encompass hardware, software, firmware, or any suitable combination thereof.

It should be understood that the above described steps of the processes of FIGS. **3** and **9** can be executed or performed in any order or sequence not limited to the order and sequence shown and described in the figures. Also, some of the above steps of the processes of FIGS. **3**, **7**, and **9** can be executed or performed substantially simultaneously where appropriate or in parallel to reduce latency and processing times.

Although the invention has been described and illustrated in the foregoing illustrative aspects, it is understood that the present disclosure has been made only by way of example, and that numerous changes in the details of implementation of the invention can be made without departing from the spirit and scope of the invention, which is limited only by the claims that follow. Features of the disclosed aspects can be combined and rearranged in various ways.

What is claimed is:

1. A method for point cloud rendering, comprising:

rendering a first image based at least in part on point cloud data, the first image corresponding to a first viewing frustum;

determining that a viewing angle has changed to a second viewing frustum;

requesting, from a memory, points from a first synthetic point cloud that are within the second viewing frustum, wherein the first synthetic point cloud has a resolution that is lower than a resolution of the point cloud data;

requesting points in a subdivision of a second synthetic point cloud that intersects the second viewing frustum, wherein the second synthetic point cloud has a resolution that is lower than a resolution of the point cloud data and higher than a resolution of the first synthetic point cloud;

reprojecting points used during rendering of the first image into a plurality of frame buffer objects of different resolutions including at least a first frame buffer object of a first resolution for points from the first synthetic point cloud, and a second frame buffer object of a second resolution for points from the second synthetic point cloud, wherein the second resolution is higher than the first resolution;

determining, for each frame buffer object, whether at least one point received from the memory corresponds to the same position in the frame buffer object as a point reprojected into the frame buffer object;