January 2020

# Virtual Reality (VR) Screenshot

Anonymous Anonymous

# Virtual Reality (VR) Screenshot

## Abstract

The present disclosure describes a method for capturing virtual reality (VR) screenshots of a three-dimensional (3D) scene for a virtual environment. The method is implemented partially on a VR application that is installed on a computing device and partially on a camera rig. The camera rig, housing one or more cameras, is connected to the computing device via a network. The one or more cameras are directed at different angles or at a specific angle to capture the 3D scene from the different angles or the specific angle, respectively. Also, the camera rig is configured to function as a stationary rig or a rotational rig. The one or more cameras capture and locally store a video of the 3D scene. The camera rig then provides the video to the VR application via the network. The VR application includes a video processing system that further includes an interpolation module, a stitching module and a capture correction module. The interpolation module receives the video and calculates motion vectors between related pixels in adjacent frames of the video. The interpolation module removes redundant frames from the video and stores only a key set of frames determined based on the motion vectors. The interpolation module further generates interpolated frames (that are not part of the key set of frames) based on the motion vectors, depending on how much precision is required. The interpolation module then sends the key set of frames, the interpolated frames, and the motion vectors to the stitching module. The stitching module interleaves the interpolated frames into the key set of frames based on the motion vectors to generate a set of stitched frames. The capture correction module adjusts the set of stitched frames to compensate for camera misalignment due to camera pose errors, and the like. The set of stitched frames, after being adjusted by the capture correction module, is used to generate an omnistereo panorama. The omnistereo panorama is used as the VR screenshot for a VR software (for example, a VR game) on a VR store.

## Problem statement

In prior art, in order to show a representation of the virtual environment for the VR software to a user, a 2D image was used. But the 2D image does not give a fair idea or enough details about the virtual environment to the user, like how his/her experience is going to be with the VR software after purchasing it. For example, a wide field of view of the virtual environment could not be presented to the user.

The present disclosure introduces a novel solution to overcome the aforementioned problem.

## System and working

The present disclosure describes a method for capturing virtual reality (VR) screenshots of a three-dimensional (3D) scene for a virtual environment. The method is implemented partially on a VR application that is installed on a computing device (for example, a mobile device) and partially on a camera rig. The camera rig, housing one or more cameras, is connected to the computing device via a network.

The camera rig captures and locally stores (for example, in a permanent or removable storage) a video of the 3D scene. The camera rig then provides the video to the VR application via the network. The VR application includes a video processing system that further includes an interpolation module, a stitching module and a capture correction module.

The interpolation module receives the video from the camera rig. The interpolation module calculates motion vectors between related pixels in adjacent frames of the video. In order to calculate the motion vector, each frame is broken down into macroblocks, where each macroblock represents a set of pixels within the frame. Also, there is very little change from one frame to a following frame in the video (usually only small movements), signifying a lot of redundancy. For this reason, instead of encoding the macroblock, a difference between two macroblocks is encoded and stored in a local storage as the motion vector. Thus, the interpolation module removes redundant frames and stores only a key set of frames determined based on the motion vectors. This way, the redundancy in the video is leveraged to achieve video compression based on the motion vectors. This reduces size of the video, and memory requirements for the video to be stored. Further, a certain level of precision may be required in the video, for example, a user may require more precise frames in-between the key set of frames. In order to achieve this, the interpolation module generates interpolated frames (that are not part of the key set of frames) based on the motion vectors, depending on how much precision is required. The interpolation module then sends the key set of frames, the interpolated frames, and the motion vectors to the stitching module.

The stitching module interleaves the interpolated frames into the key set of frames. The interleaving includes the stitching module stitching together the interpolated frames and the key set of frames based on the motion vectors to generate a set of stitched frames.

The capture correction module is configured to adjust the set of stitched frames to compensate for camera misalignment due to camera pose errors, and the like. For example, if the camera pose errors (like, errors in the estimation of camera parameters, such as, a position and an orientation of the camera) occur when the camera rig captures the video, the capture correction module blends two or more columns of pixels from several frames in the set of stitched frames to remove artifacts. The set of stitched frames, after being adjusted by the capture correction module is used to generate an omnistereo panorama, which is used as the VR screenshot for a VR software (for example, a VR game) on a VR store. The user may then view the 3D scene using a head-mounted display (HMD), and the omnistereo panorama makes it possible to show a wide field of view of the virtual environment to the user. Thus, the user is able to get a preview/demo of the virtual environment before purchasing the VR software.

## Additional embodiments

In an additional embodiment, the camera rig is embedded in at least a portion of the computing device. For example, the camera rig is embedded in the mobile device on which the VR application is installed. The VR application has a direct access to a permanent or removable storage of the mobile device. The VR application may directly access the video of the 3D scene captured by the one or more cameras.

In yet another embodiment, the camera rig is configured to function as a stationary rig. In such a configuration, the multiple cameras housed on the camera rig are utilized to capture additional outward angles of view for the 3D scene in a stationary position.

In yet another embodiment, the camera rig is configured to function as a rotational rig. The camera rig is configured to rotate around 360 degrees to sweep and capture all or a portion of a 360-degree view of the 3D scene.

In yet another embodiment, the camera rig is configured to capture particular angles of the 3D scene. For example, the one or more cameras, housed on the camera rig, are directed at a specific angle. All or at least a portion of the 3D panoramic content captured from that angle is then to be processed to generate the omnistereo panorama of the 3D scene.

In a yet another embodiment, the one or more cameras are directed at different angles to capture the 3D scene from different angles.

## Conclusion

Playing virtual reality (VR) games has become quite a norm these days. Not just games, the VR has got immense potential to contribute to education, training, healthcare, simulation, and much more. A lot of research and development goes into the designing of VR software intended for these applications. The VR software should be designed for the capabilities, convenience and flexibility of the user. To leap a step forward in this direction, the VR software should provide an option to the user to get a preview/demo of the virtual environment before purchasing the VR software. The present disclosure provides a mechanism to provide a snippet/screenshot of the VR environment that the user is going to experience after purchasing the VR software.