Depthmap based Stereoscopic Panorama Viewer for Head-Mounted Displays

Helmut Dersch

November 10, 2016

Abstract

PTViewer3d is an experimental spherical panorama viewer that generates stereoscopic views on the fly from monoscopic panoramic images and depth maps. Unlike conventional two-image based viewers it provides stereoscopic vision in all directions including vertically up, down and z-rotated views, as well as head-motion parallax. This is a port of the previous desktop version (Windows and MacOS) for GearVR/ Oculus and for Google Cardboard.

Contents

Figure 1: Input images for stereoscopic viewer.

1 Introduction

Depth maps are images whose pixel values represent a measure of the distance of scene objects to the viewer. Stereoscopic views can be rendered from monoscopic images and depth maps by displacing pixels a distance-dependent amount and filling the gaps by some interpolation mechanisms. This is established and will not be reviewed here. Also, the creation of panoramic depthmaps is not subject of this article, an image based method is described here [1].

Rendering of such views used to be a timeconsuming operation. Progress in graphics hardware and development of simplified algorithms now makes it feasible to realize interactive displays. Recently, we published a desktop version of such a viewer, PTViewer3d, which renders 60 frames per second on 2k-displays requiring just a modest current graphics card (see here [2]).

Conventional stereoscopic viewers, whether panoramic or not, use two source images and present those to the two eyes of the viewer. Viewing parameters are then locked to the parameters of the stereo camera which took the images. Any movement of the head, be it lateral or rotational, poisons the stereoscopic experience. As an example, the viewer may not turn his head around the viewing direction. Even looking to the left, right, tilt up or down in a wide angle view damages the stereoscopic effect. This becomes devastating in 360 degree immersive panoramas where views vertically up and down are permitted, but incompatible with stereoscopy. The depthmap-based approach generates geometrically correct views under any circumstance, even slight lateral movements (parallax) can be simulated. The main limitations are occlusions in the scene which the algorithm has to interpolate.

This article describes a version of PTViewer3d running on head-mounted displays. Versions for GearVR/ Oculus [8] (`ptviewer3d.apk`) and Google Cardboard [9] (`ptviewer3dc.apk`) are available for download. They are identical regarding the viewer engine.

2 Requirements

PViewer3d runs on Android version 6 (Marhmellow) [10] and uses OpenGL [6] version 3.1 and OpenCL [5] version 1.1 to render views. All GearVR capable Samsung phones with Adreno GPU should be suitable for both GearVR and cardboard versions. Because of the tedious GearVR installation procedure it might be easier for GearVR-testers to start with the cardboard version and use one of the cardboard enablers for GearVR available from Google Play. In the long run, however, it is definitely worth switching to the much smoother GearVR-version. The performance (image quality, render speed) is almost identical for the two versions, but the cardboard-app looks jerky due to not using the fast position sensors in the GearVR.

Non-GearVR phones should work if they support the mentioned features (versions for Android, OpenGL, OpenCL). Of these, OpenCL may be problematic since it is not official part of Android. The availability of OpenCL can be tested with this app [7]. Our application is linked against the OpenCL library supplied with Adreno GPUs, phones based on this chip (Adreno 420 or better) are likely to run PTViewer3d. Phones with GPUs from other vendors may require additional library files or may not work.

3 Installation

Both versions of PTViewer3d are installed like any program on Android from "unknown sources". By default, this option is disabled for security reasons, and has to be enabled before downloading the application to the phone. This can be done in the preferences settings, see this link [11]. Then, the application file can either be directly downloaded via download link to the phone, or transferred from the desktop computer to the phone via USB and then opened by the file explorer on the phone. Both procedures will start the installation process.

Unfortunately, the GearVR-application only runs if it is signed for the user's phone. This can be done in two ways: Either the user sends the phones hardware

serial number as described here [11] to the developers at der (at) hs-furtwangen.de. Future versions of the program will be automatically signed for this number. Or the user downloads a custom signed version from the service "SideLoadVR" [12].

To access panorama images on the phone the user has to grant access to storage in the application manager. The application requires internet access to load remote images and html files. Internet access only occurs when the user opens the "home" link in the built-in file browser, see usage below.

4 Usage

Both, Cardboard and GearVR versions provide the same user interface, the only difference being that the GearVR-version also responds to swipes of the built-in touchpad. The main means of interaction, though, is tapping. After starting the application and inserting the phone into the frame, the built-in example panorama "Granite" and the main dialog are displayed. If the dialog is out of sight tapping twice makes it reappear in the viewers direction.

Dialog windows are rendered as overlays over the current panorama in the direction the user hit tap. Dialog windows are accompanied by a visible cursor (red circle) used for making selections. Tapping outside a dialog window closes the dialog. Moving the cursor over selectable items (blue) changes their color (green). While green, tapping activates the selection. Pure text windows and alerts are closed by tapping anywhere.

While dialogs are present, the panoramic image is displayed in monoscopic mode. Closing the main dialog (either by selecting "Close" in the dialog, or tapping outside the window) switches to stereoscopic view, unless this has been turned off in the previous dialog. In the default "Granite"-panorama, immediately a cave-like 3-d structure becomes visible. Swiping (GearVR) makes this more apparent or choosing "Wiggle" from the main menu (both versions). It should be mentioned, that the "Granite"-panorama is actually generated from textures obtained from polished flat granite, and an artificial random depthmap.

4.1 Main Dialog

The main dialog is presented upon startup, and whenever the user taps into the panoramic view. The following options are available:

Open: Displays filedialog for loading files from the device or from the internet, see below.

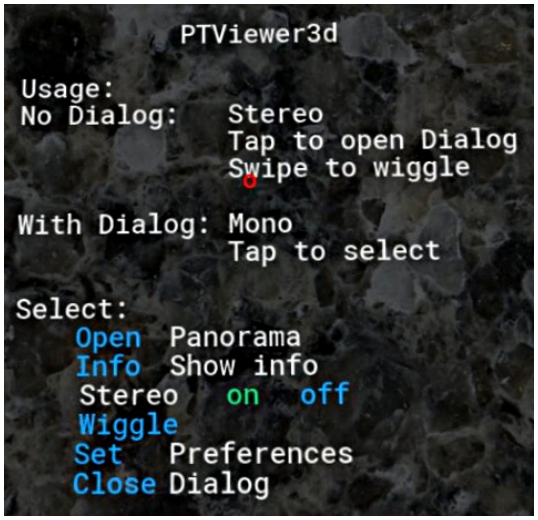

Figure 2: Main Dialog.

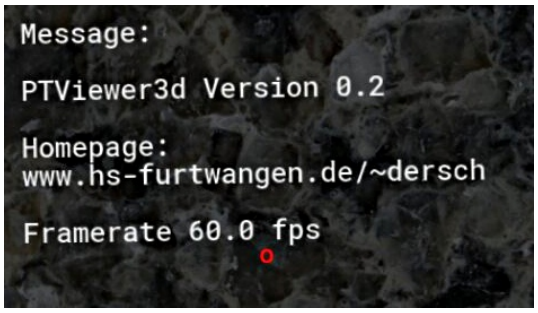

Figure 3: Info Dialog

Info: Display Information, homepage, and current render speed. The speed is the number of rendered frames (per eye) counted during the last second before opening the dialog.

Stereo on/off: Turn stereo engine on or off. This overrides the automatic stereo detection mechanism. If PTViewer3d discovers a depthmap, stereo is automatically turned on, otherwise off.

Wiggle: Simulates a lateral circular head motion (parallax), stereo must be "on". The amplitude of this movement can be set in the Preferences dialog.

Preferences: Opens the Settings dialog, see below.

Figure 4: Open File Dialog.

4.2 Open Dialog

Filedialog for loading image and html-files. If granted permission, PTViewer3d loads files from these locations

- Device storage and SDCard. Android file access is complicated since many parts of the filesystem are not accessible, and filestructures are mapped to names in strange patterns. PTViewer3d displays two locations: The directory named by the Android system variable `ANDROID_STORAGE` is mapped to the name `/sdcard`, and `EXTERNAL_STORAGE` to `/storage`. On my phone, the removable sdcard then appears at `/storage/9C33-6BBD`; this name may be different on other phones.

- Assets are files located inside the application bundle. The content of this asset directory is also displayed and comprises the default "Granite" panorama.

- Internet access is available through the "home" link which loads the file `http://www.hs-furtwangen.de/~dersch/PTViewer3d/index.html` and displays links in it. The URL representing "home" can be changed by the user, see below.

Image loading is handled by `stb_image` [4]. These formats are supported (from the documentation):

JPEG baseline and progressive (12 bpc/arithmetic not supported, same as stock IJG lib)

PNG 1/2/4/8-bit-per-channel (16 bpc not supported)

TGA (not sure what subset, if a subset)

BMP non-1bpp, non-RLE

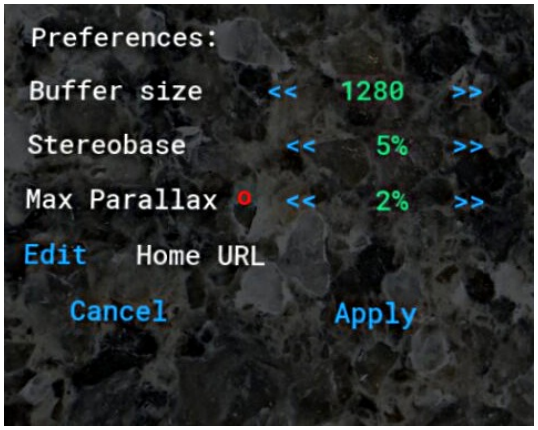

Figure 5: Set Preferences Dialog.

PSD (composited view only, no extra channels)

GIF (*comp always reports as 4-channel)

HDR (radiance rgbE format)

PIC (Softimage PIC)

PNM (PPM and PGM binary only)

Images are loaded into memory if imagewidth is twice the height. If the image has 4 channels (rgba) or if another image with the same basename extended by `_depth.png` is present, a stereoscopic view is displayed, otherwise stereo is turned "off". There is a built-in limit on image-size set by the maximum texture-size of the OpenGL implementation. On the Samsung Note 4, this value is 16384 pixel width.

HTML-files can also be loaded. All image links and links to other HTML-files found inside this file are presented as a directory structure. For an example see or download the default homepage
`http://www.hs-furtwangen.de/~dersch/PTViewer3d/index.html`.

4.3 Preferences

Various settings can be read and changed in this dialog. Closing this dialog using the "Apply" option makes the changes permanent, and saves them to a preferences

file. The settings can be reverted to the default values by choosing "delete files" in the application manager.

Buffersize: refers to the width of the per-eye framebuffer used to render images. To avoid degradation due to sampling this should be as large or larger than half of the screen width, e.g. 1280 for the Samsung Note 4. However, render speed is largely affected by this value, and may be too low at this setting. For the Samsung Note 4, I get 40 frames per second for buffer size up to 2048 pixels in monoscopic mode, but for just 1024 in stereo mode.

The GearVR-system suggests a buffer size of only 1024 pixels, mainly to provide speed for gamers. Significant improvement of image quality at the expense of responsiveness can be obtained by increasing this value, which makes the program useful as replacement for the standard monoscopic panoramaviewer.

Stereobase: in percent of eye-buffer width. This value determines how pixel values in the depthmap are translated to displacements in the final view. Distant scene objects with large depthmap values (white regions in the depthmap) are not displaced, and rendered in left and right view alike. Close objects (black regions in the depthmap) are displaced by half of the stereobase to the left in the right image, and to the right in the left image. Changing stereobase scales the perceived depth impression. The optimum value depends on how the depthmap was created.

Parallax: in percent of eye-buffer width. This value sets the maximum lateral displacement for the "Wiggle"-effect (see Main menu above) and for the lateral movement caused by swiping the touchpad in the GearVR version.

Edit Home URL: opens a textinput dialog which allows the user to change the URL for the "home" link in the filebrowser.

5 Some Experiences

Best results are obtained with smoothly varying depthmaps like the "Granite"-panorama. In this case no occlusions exist, and no interpolations are required. The "LakeTitisee"-panorama from the homepage is more problematic since it contains

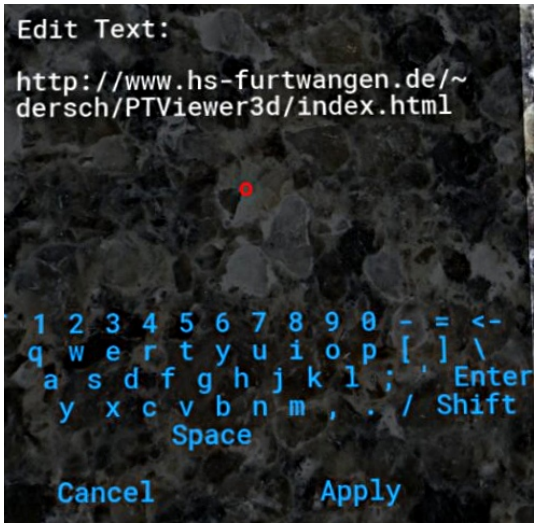

Figure 6: Edit URL.

abrupt depth changes, e.g. at the tree-trunks. The edges exhibit some artefacts. Regions without such changes look much better.

Also, there are a couple of errors due to imperfections in the depthmap. Depthmap creation is not trivial, an image based experimental solution is described here [3].

References

[1] depthmap.exe, a small utility to create depthmaps from pairs of images.

[2] H. Dersch Homepage

[3] Adding Depth to Panorama Viewers.

[4] https://github.com/nothings/stb

[5] https://www.khronos.org/opencl/

[6] https://www.opengl.org/

[7] OpenCL-Z on Google Play.

[8] GearVR/ Oculus.

[9] Google Cardboard.

[10] Android Marshmellow.

[11] How to get Samsung Device ID.

[12] SideloadVR.