Eye Alignment Display (Quarter Report) **(Draft)**

The project focuses on using light field displays to create alignment cues for the patient to aid in eye diagnosis where precise eye alignment is crucial, ex. In retinal imaging. This was divided into three broad categories –

1. Graphics Programming: Generating the multiview / light field image which will aid in 3D depth perception of the image contents when viewed from a Light Field Display. Further models to simulate how the image appears on the retina of the viewer.
2. Building the hardware required for this. Creating a light field display with the help of a screen and appropriate pinhole mask. This pinhole mask helps in giving the perception of parallax while viewing a specially generated multiview image of a scene.
3. The final head mounted hardware which can be worn where all this different parts fit together and perform the required test as required.

Theory:

The theory required to understand this is very minimal, at least, to what is given here in the report. Our method generating parallax relies on the following two steps:

1. Proper alignment between the mask and the image displayed on the phone. For this it is essential that we get the mask printed with precise dimensions including the dimensions of each pinhole so that it correlates exactly with the phone PPI.
2. Generation of the multiview image keeping in mind the calculations done while producing the mask and knowing how this image will be displayed on the phone display.

What determines the pinhole mask dimensions and how is it related to the phone PPI?

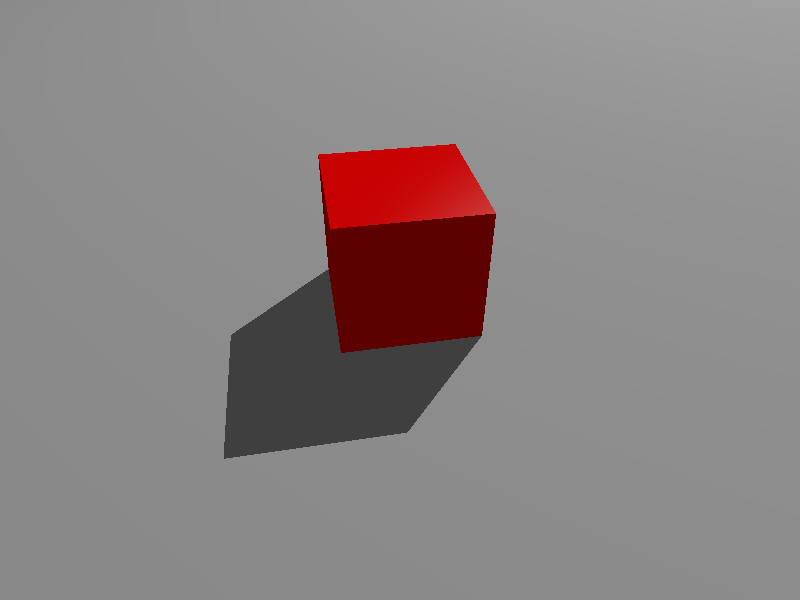
One basic way to generate parallax is to make the viewer see different set of images as he moves relative to the screen. In order to do this, we need to first restrict his field of view of the scene so that he only sees one part of it. This is where the pinhole masks come into picture. The pinhole masks creates a restriction to the field of view of the screen and hence you only see a part of it depending on the size of this pinhole relative to the screen. This pinhole mask will rest on your phone display which already has a glass screen which acts a separator and results in a gap between the screen and the pinhole mask. This means that now, if you move your head while looking through this pinhole, you will see different parts of the screen under the pinhole mask. This is analogous to looking through a keyhole into a room. Now, ideally we should only be looking at one pixel at one instant and our head movement should enable us to view another pixel on the screen in the opposite direction to the movement of the head. Hence, the pinhole mask should coincide with only one pixel in a group of them to get the best parallax.

How is the multiview image formed and which pixels should coincide and which ones should not?

A multiview image is formed by compressing the 4D light field on a scene in 2D. This is achieved by forming super pixels (group of pixels) which contain information about the same spatial point in the scene but they have differnet angular data. Simply put, we take different angle views of the same point in the scene and arrange them together in an array to form a super pixel at the same point to which this point correlates to. This is done for all the points in the scene and hence, we obtain the multiview image. The pinhole mask should coincide exactly with this superpixel allowing it to view only on pixel when viewed at a certain angle and revealing another pixel when the head is moved relative to the screen. The pinhole should match exactly with the center of this superpixel. For our experiments, we are using a 5x5 superpixel array allowing us 5 different horizontal and vertical views which can be seen by tilting our head with respect to the screen.

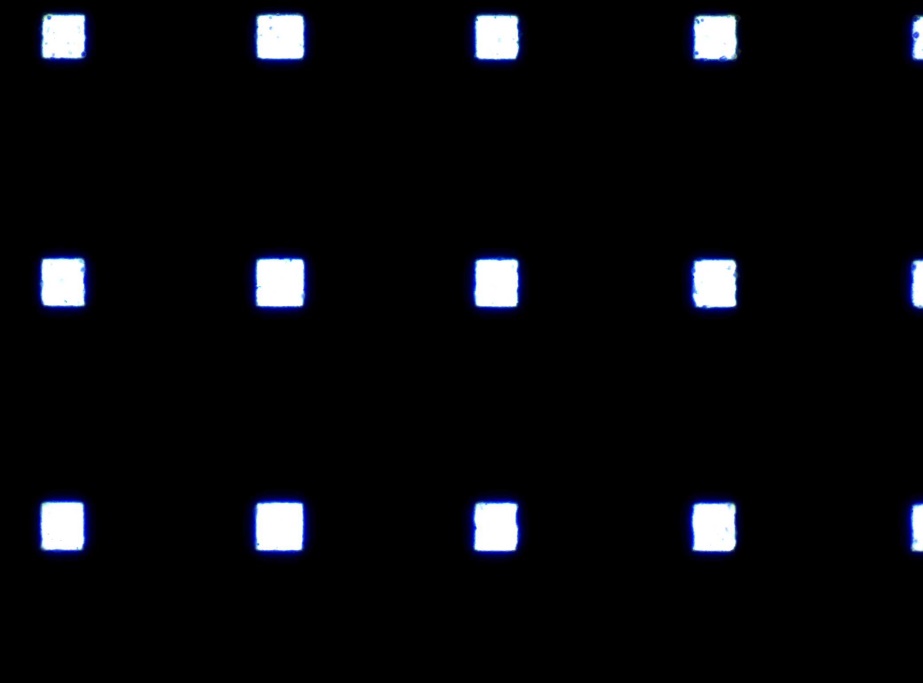
Progress:

1. Graphics Programming: We started with generating a simple image with a Red Box on a grey background with shadows in a ray-tracing software, POV-Ray. This was then modified to generate a set of these images in sequence with the cameras moving +/- small distances in order to generate images from different perspectives. Once this was completed, we moved forward with writing MATLAB scripts for easy conversion of these various images into a single Light Field Image and then a Multiview Image.



A simple image generated in POV-Ray

1. Parallely, we stared looking at different phone displays and generating the pinhole mask for this. The mask given in the original source we were following for creating the light field display   
   <**http://displayblocks.org/diycompressivedisplays/parallax-barrier-display/ >** suggested a mask which needed to be printed with a 5050 dpi printer which was not possible nearby in Hyderabad. Hence, we modified the mask to work with Videocon A55HD and Moto G, by printing a 600dpi version of it. After a few days of negotiation, we received the first batch of these masks. They were good, had good continuity meaning that we were on the right path in terms the “technology” used for printing. We were lithography, same used for generating films for PCBs for this. However, the mask we received had an offset in the width and the pinholes were manufactured as rectangles instead of squares. The result when viewed through Moto G phone was something like this. The image we earlier generated using the Pov-Ray software did not give any exciting result. We then tried to use the images given on the displayblocks site (mentioned above) and then were able to generate sufficient parallax. This meant there was something wrong in the way we were generating our images.



Rectangular pinholes instead of square.

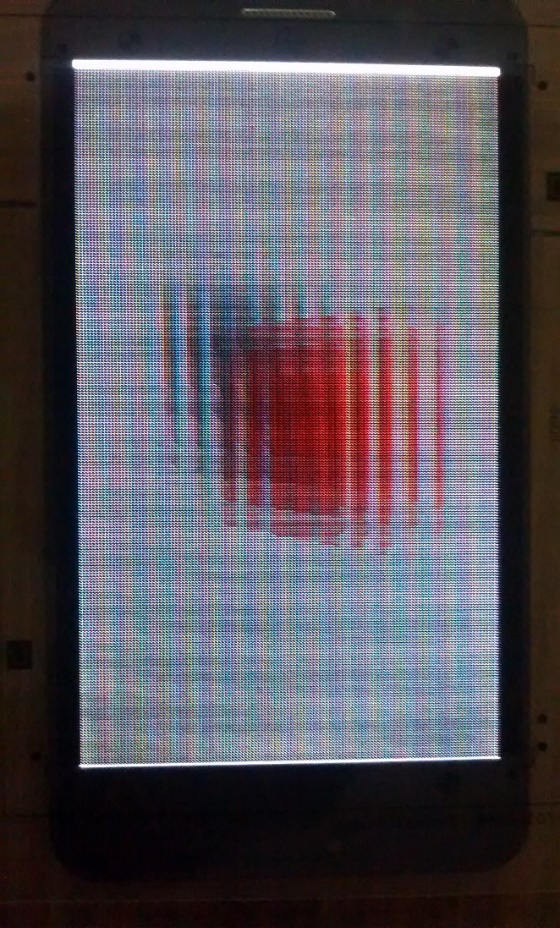
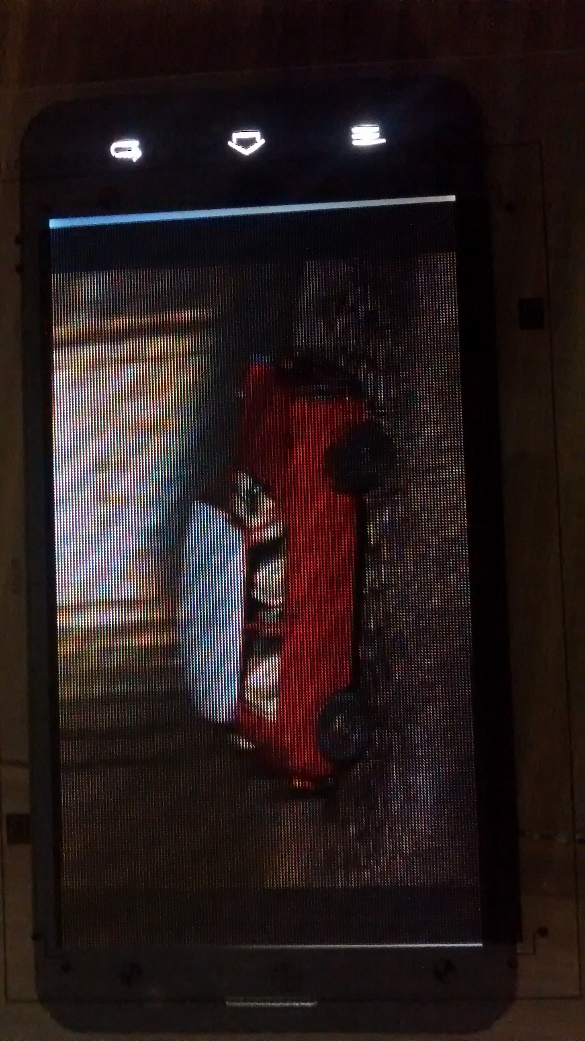
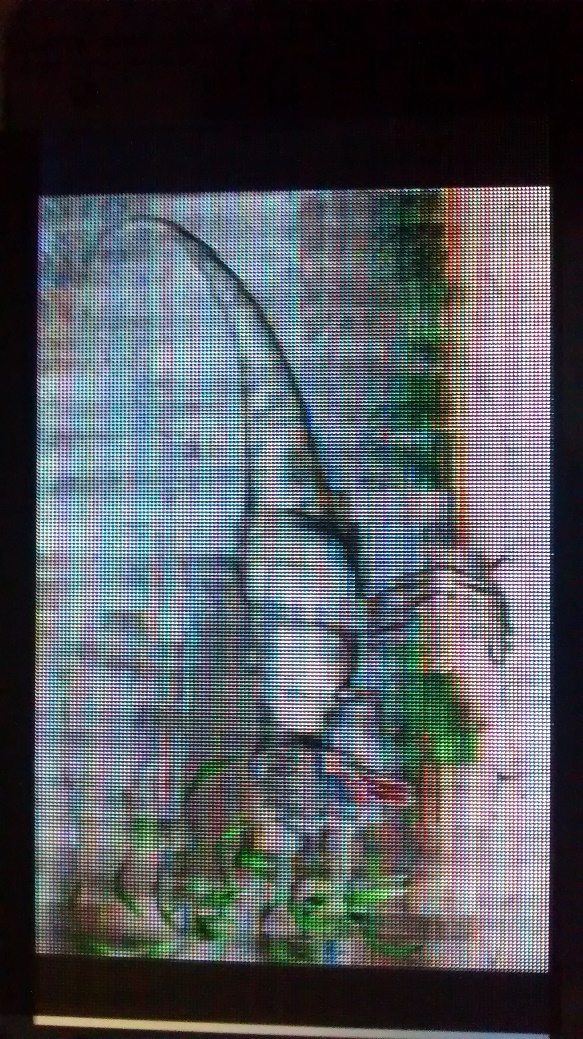


Image generated form Pov-Ray when viewed with the mask.



Images downloaded from the site when viewed with the mask.

1. We soon realized something was wrong as the image was showing no parallax at all! We then realized that the image we had been generating assumed that the camera moves linearly in the horizontal and vertical direction with the point at which it was viewing as moving laterally with the camera. We then modified it to keep its gaze fixed at a particular point while moving the camera in a sphere at a distance ‘d’ with small angle deviation ‘α’. We then checked with this image on the printed mask using Moto G phone. The result was better but far from perfect. We could see a lot of jumping in the scene above a certain ‘α’ and then no movement below that angle deviation ‘α’. This jumping was due to mismatch between the pinhole mask and the screen pixel locations.
2. Next, we received the masks ordered by Tristan which were made from a 5050 dpi image (so we thought). We started testing with that only to realize that even this was not matching perfectly with the phone and the parallax was not perfect. While viewing this mask through a microscope we realized that the mask pinhole array wasn’t as fine as it should have been if it were a 5050 dpi printed mask. So, we found out that even this was a 300 dpi mask. While viewing the images generated for this mask we found better parallax than the previous mask as these pinholes were squares instead of rectangles.
3. We got a final set of masks made from the local vendor. The dimensions were given according to the 5050 dpi image. We are yet to test those masks.

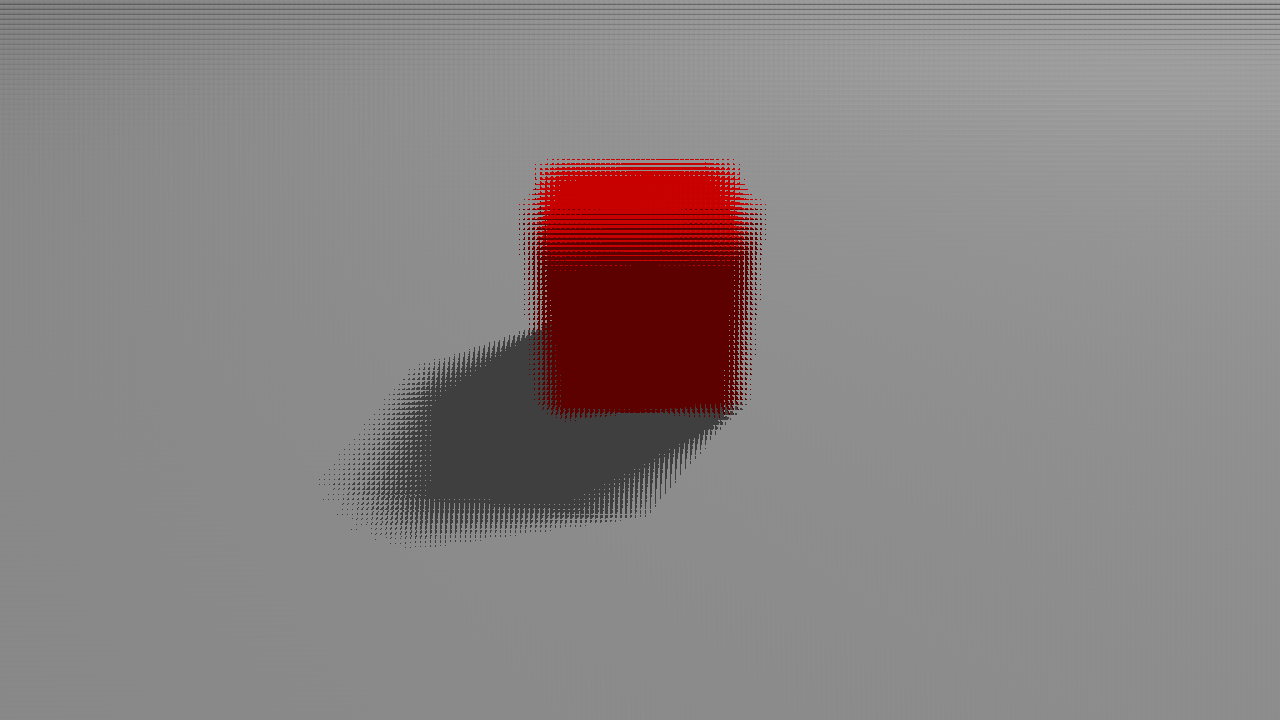
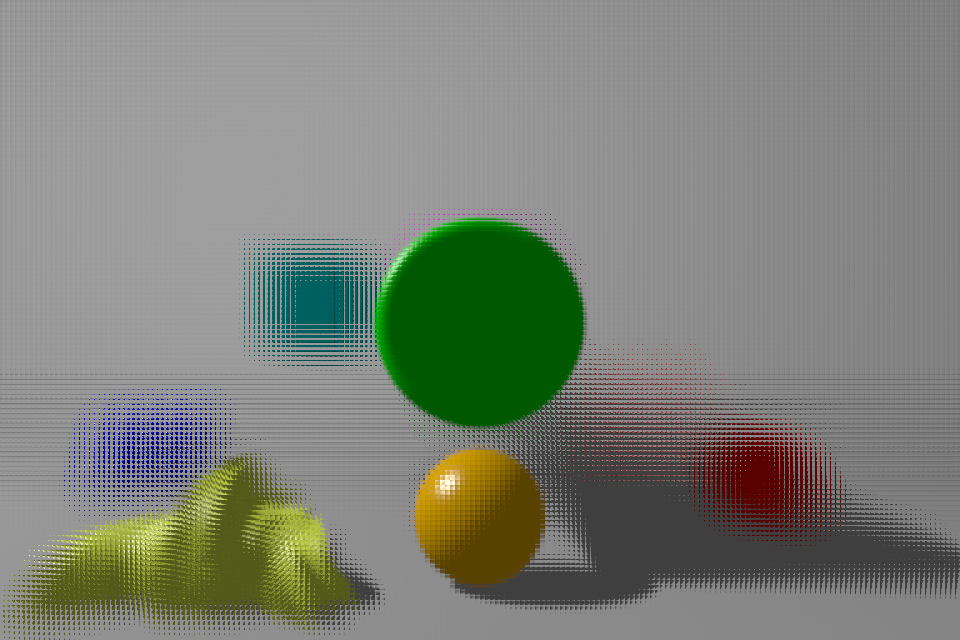
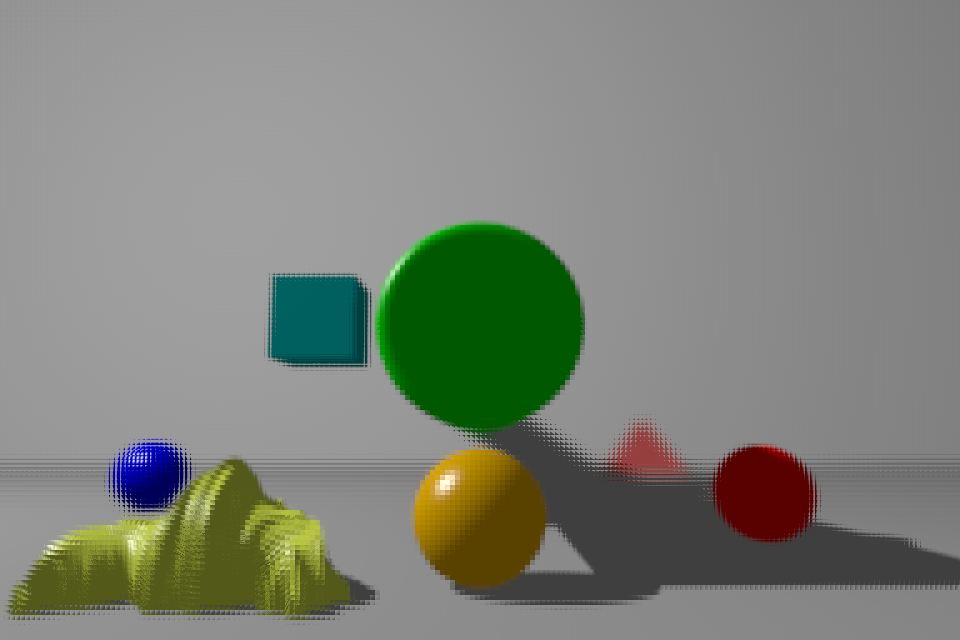


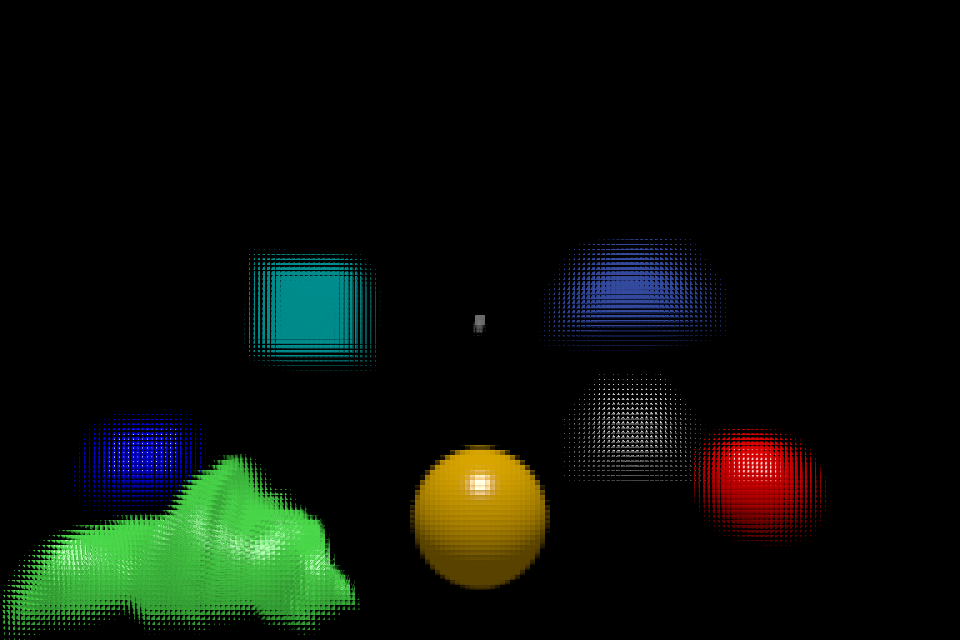
Image generated by moving about a spherical locus.



Debug Scene, Large α



Debug Scene, Small α



Final Debug scene, Large α



Final Debug scene, Small α