Holojackets

Semester Update: Spring 2018

April 23, 2018

Georgia Tech Vertically Integrated Projects

Georgia Tech Research Institute

Agenda

- Team Focus
- Problem Statement
- Team Structure
  - Progress
  - Conclusion/Project Plan

Team Focus

- Prototype 3D displays that allow viewing from any orientation without requiring special artifacts.

- Utilize light field displays that will allow viewers to naturally perceive 3D scenes with normal visual cues such as parallax, vergence, and focus accommodation.

Problem Statement

- End goal is to use 3D image information from light field camera to create an image on computer screen which you can look at from different angles to see different views.

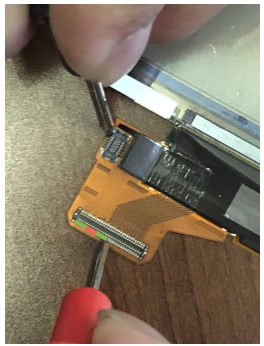
HoloJackets Sub-teams

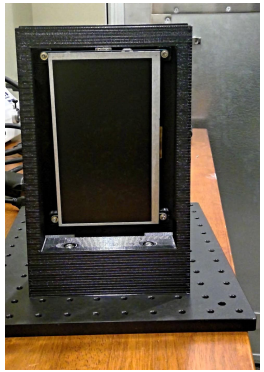
Images
Image Capture

Electronics
High Resolution Display Group

Display
Parallax Barrier

Lens
Micro-Optics





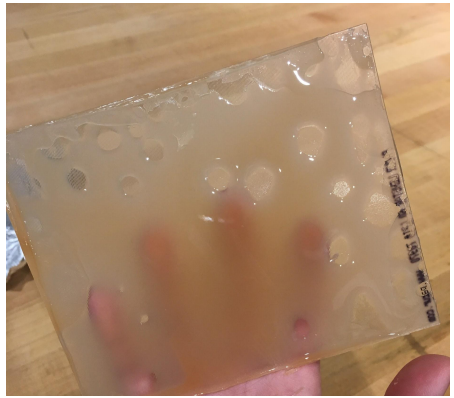


Image Capture: Purpose

- Previously attempted to create original image data that could be used when prototyping 3D displays
  - Pictures were taken using a slider bar and manually measuring the distance the camera needed to move which was subject to human error
- A motorized gantry was built to eliminate error and gather a precisely measured set of images
- Our goals:
  - Use a Raspberry Pi and Pi camera mounted to the gantry
  - Control the motors with an Arduino
  - Simultaneously run the motors and take photos

Image Capture: Background Information

Programs used: Raspbian, Python, GCode Sender

Concepts Used: Embedded Linux

Materials Used: Raspberry Pi, Raspberry Pi Camera Module, Arduino, CNC Shield, DC Motors, Limit Switches

Image Capture: Research

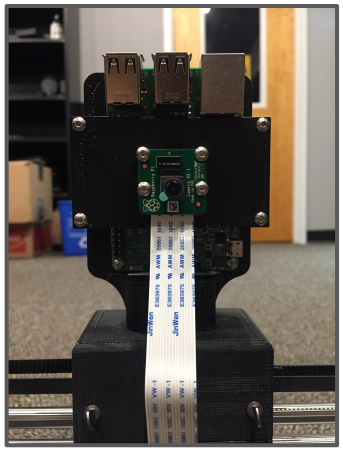
- Taking continuous pictures with a Raspberry Pi Camera
  - Cleaner layering of photos on Photoshop

- Using G-Code to move the gantry
  - Limit Switches

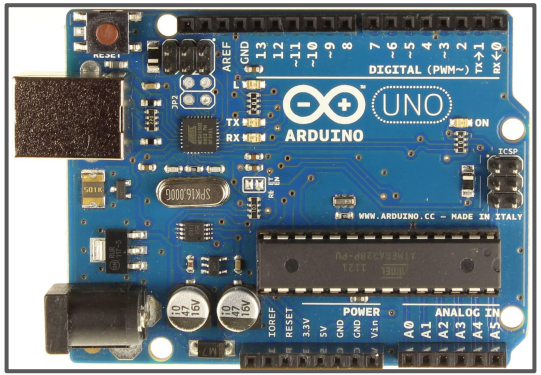
Image Capture: Hardware Used

Motorized Gantry (pictured to the left)

Raspberry Pi (pictured to the left)



Raspberry Pi Camera Module (pictured to the right)



Arduino Uno (pictured to the right)

Image Capture: Progress

- Authored Python code for the Raspberry Pi to take a photo using the Pi Camera when GPIO 17 is high

```
# /usr/bin/python
# -*- coding:utf-8 -*-
import RPi.GPIO as GPIO
import datetime
import time
import os
from picamera import PiCamera
from time import sleep

buttonPin = 17
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(buttonPin,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
#GPIO.output(buttonPin, GPIO.HIGH)
#GPIO.output(buttonPin, GPIO.LOW)
#GPIO.output(buttonPin, GPIO.HIGH)

camera = PiCamera()
```

```
date = datetime.datetime.now().strftime("%m_%d_%Y_%H_%M_%S")
os.makedirs(date)
os.chdir(date)

pictureCount = 0
prev_input = 0
```

```
while True:
    sleep(0.05)
    input = GPIO.input(buttonPin)
    #print(str(input))
    if ((not prev_input)and input):
        name = 'image' + str(pictureCount)+'.jpg'
        pictureCount+=1
        camera.capture(name)
    prev_input = input
```

Key Logic

- Sent commands using Universal Gcode Sender (UGS) to the Arduino that were able to run the motors and move the mount to the specified location
  - X 10 -- move to X position 10
  - Y -20 -- move to Y Position -20

Image Capture: Obstacles Encountered

1. Synchronization of code between the Raspberry Pi Camera Module and the Universal GCode Sender via the Arduino

2. Raspberry Pi overheating

3. Gantry needs calibration before each use

4. Limit switches suffering from communication interference

Image Capture: Goals Moving Forward

- 1.) Create Arduino code to allow the horizontal motors and switches to synchronize with the pin output used for capturing the images on the Raspberry Pi Camera
- 2.) Find an alternative method to run Python code on the Raspberry Pi to control both the motors and the picamera, removing the need for an Arduino (more difficult to pursue)
- 3.) Create an official list of instructions for calibrating the gantry.
- 4.) Correct the communication errors between the limit switches.

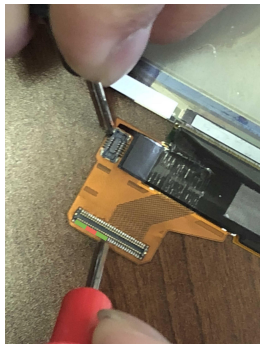
HoloJackets Sub-teams

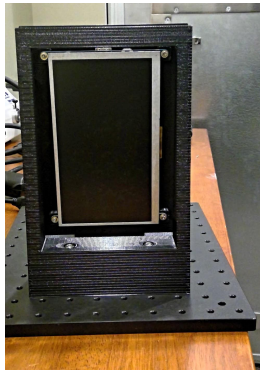
Images
Image Capture

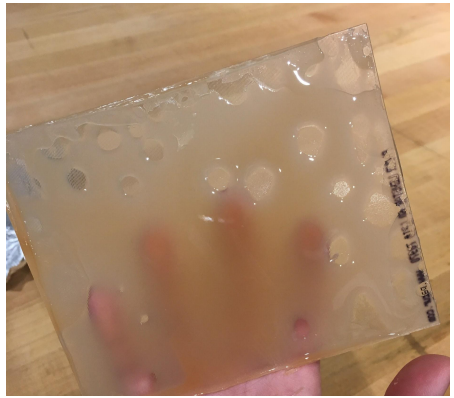
Electronics
High Resolution Display Group

Display
Parallax Barrier

Lens
Micro-Optics







High Resolution Display Group

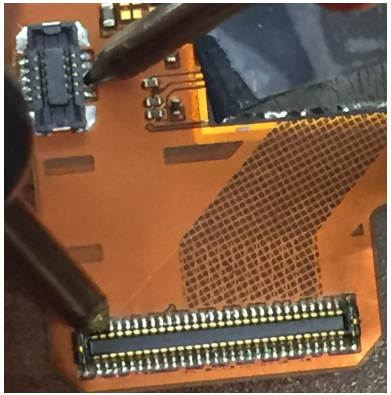
- Goal: To use intrinsyc board interface with the Sony Xperia Z3 Premium display
- Problem: intrinsyc board does not naturally interface with phone screens
- Purpose: Creating an PCB interface between the intrinsyc board and Sony Xperia Z3 Premium display

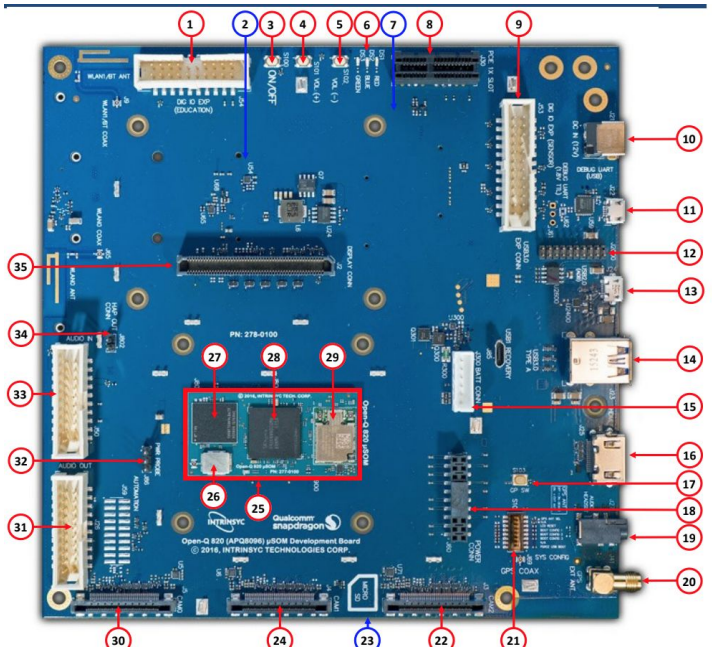
New Challenger Approaching

- This team was created to be the link between the parallax barrier group and the algorithms group.
- Instead of loading information directly into the screen, code can be loaded to the intrinsyc board.
- This board would communicate through the custom PCB to display the prompted images.

High Resolution Display Group: Background

- Had to gather all the necessary hardware (the carrier board, phone screens, and multimeter).
- Determine how the 10 pins on the LCD screen are used for display with which of the 100 pins on the carrier board.
- Used the multimeter to test pins connections.





Background Information

Programs used:ImageJ, Eagle, Kaicad

Concepts Used: Working knowledge of PCBs and how information is transferred via electronics, Voltage, Resistance

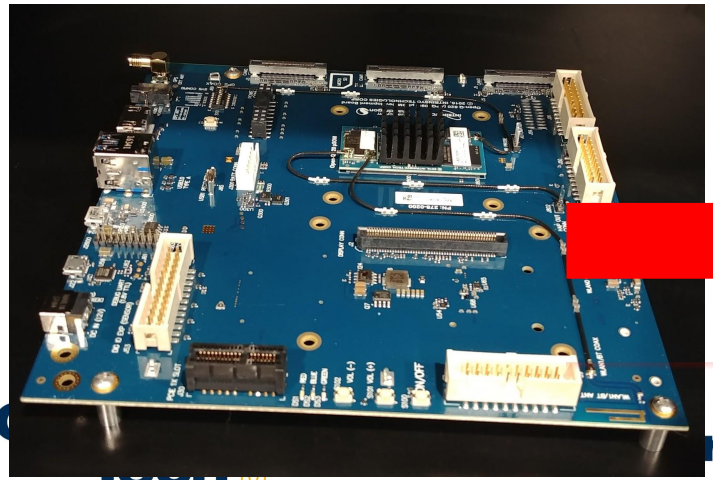
Materials Used: Intrinsyc Pin Layout, Sony Xperia Pin Layout, SD Card

Progress

- Identified the sets of connected pins on the Intrinsyc MIPI DSI connector and LS055D1SX04 screen.
- Identified need for PCB to interlace with both the intrinsyc board and the Sony Xperia screen
- Identified which kernel is compatible with the carrier board.
- Identified that the 820 and µ820 boards are similar if not identical in pinout

Path Forward

- Design PCB using EAGLE
- Print the PCB using the equipment in the Van Leer ECE lab that will be opening in the Fall of 2018.
- Confirm pin layouts for Intinsync board and Sony Xperia Display



Custom PCB

rated Projects

Conclusions

Ideally, we would be able to create a display interface between the phone screen, main board, and an intermediate board to connect the two. We'd then be able to program the board to display and manipulate images.

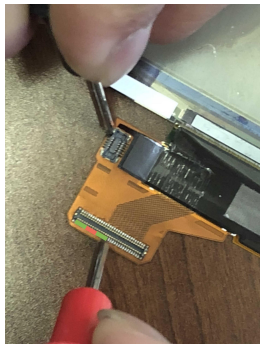
HoloJackets Sub-teams

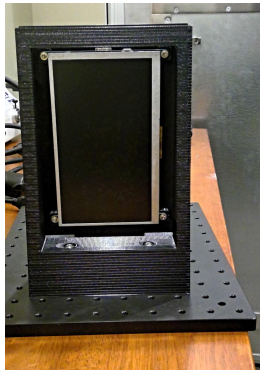
Images
Image Capture

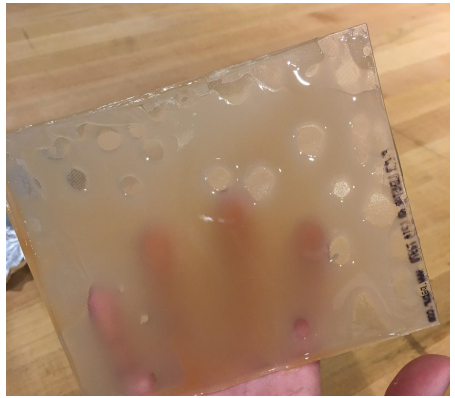
Electronics
High Resolution Display Group

Display
Parallax Barrier

Lens
Micro-Optics





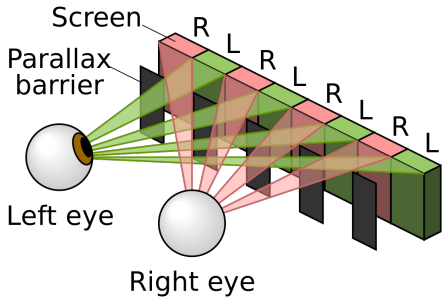


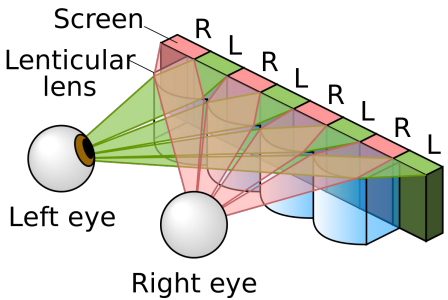
Parallax Barrier/Algorithms Subgroup Overview

- Background
- Progress
  - Display
  - Raspberry Pi
  - Algorithms
  - Printer Calibration
    - Parallax Barrier Templates
    - Two-color Experiment
- Next steps
  - Short term goals
  - Long term goals

Background: What are Parallax Barriers?

- The parallax barrier works by specific blocking parts of the image from being viewed at specific angles
- Blocking different parts of the image at different viewing angles allows for the single image to have multiple images
- This can allow for 3-dimensional images by providing different views to the right and left eye
- It also allows for a single image to show multiple images





Progress: Display

- We manually tried to measure the pixel dimensions by:
  - making the display show a 1024x600 black and white striped image of 40 pixel width white and black stripes
  - Measuring the width of each 40 pixel line with a ruler
  - Dividing the obtained width by 40
  - The manually measured dimensions for the pixels were 170.33 ppi (pixels per inch)
- The official specs (found on the manufacturer's website) were 169.55 ppi

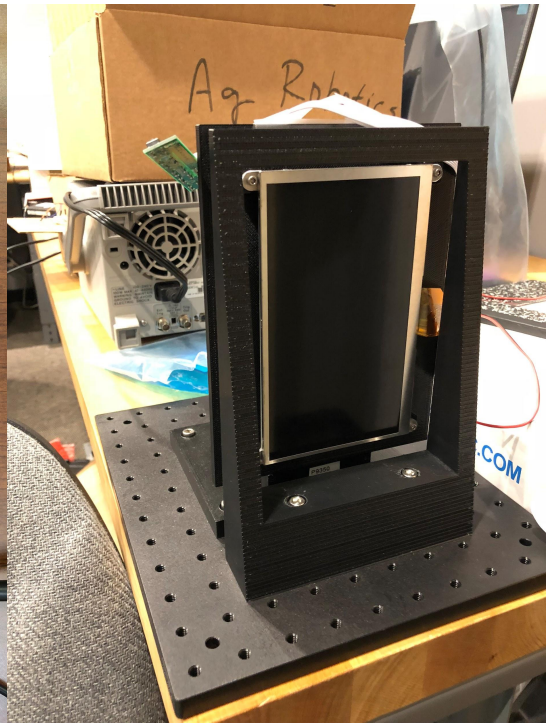
Progress: Raspberry Pi

Operating System:

- Raspbian: own version of Linux

Displaying images:

- Using bluetooth/GT other to display templates
- Eliminates need for scaling images onto the display



Progress: Raspberry Pi

Benefits:

- Consistent inputs to the display (os, screen resolution)
  - A mounted raspberry pi eliminates possible errors when using various sources of input (laptops)
- Smaller form factor allows ease of handling
  - the only changes to the system physically will be mounting the barrier and an sd card

We obtained a fresh raspberry pi that will be dedicated to the parallax barrier research. Once mounted with an OS, we will always have a reliable mean of displaying images on the LCD.

Progress: Algorithms

- Made 2 small programs in Java to simplify the parallax barrier and image creation process
- Development:
  - Started by using the graphics of JFrame, but it would only create a temporary image
  - Figured out how to use Graphics, BufferedImage, and write() to save the created image
  - Working on conversion to Python since it is the native language for the Raspberry Pi
- Functionality:
  - BlackStripe and ImageStriper take in parameters using Scanner and command prompt/terminal
  - BlackStripe asks for the black/white pixel widths, overall image dimensions, horizontal or vertical stripes, and how much to scale the image by
  - ImageStriper takes in two images, stripe width, and horizontal or vertical stripes

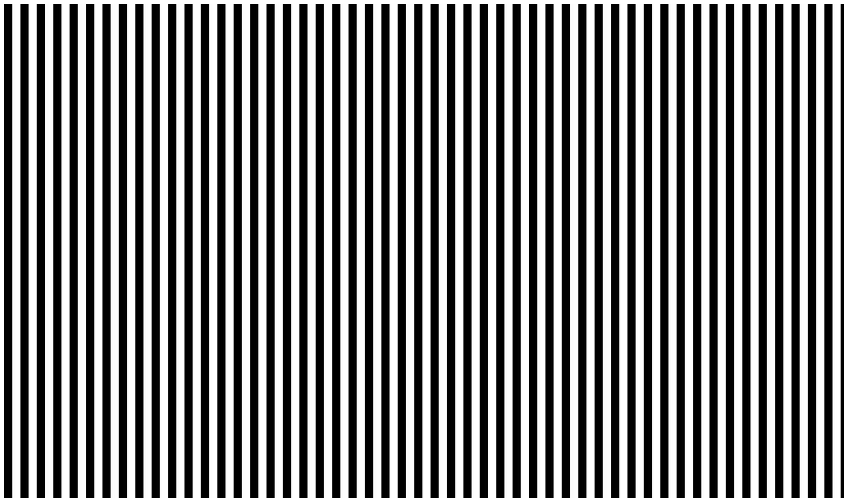
Progress: Algorithms

- Limitations:
  - BlackStripe makes the same dimensioned stripes across the entire output image, working on adding functionality to be able to apply functions that can dilate/shrink stripes across the image
  - ImageStriper is only able to take in two images that must be of the same dimension, anything else would cause the program to crash
  - ImageStriper similarly has only constant striping across the output image, working on a similar function as BlackStripe to dilate/shrink stripes across the image

Sample Outputs of BlackStripe

Sample Output of BlackStripe with:

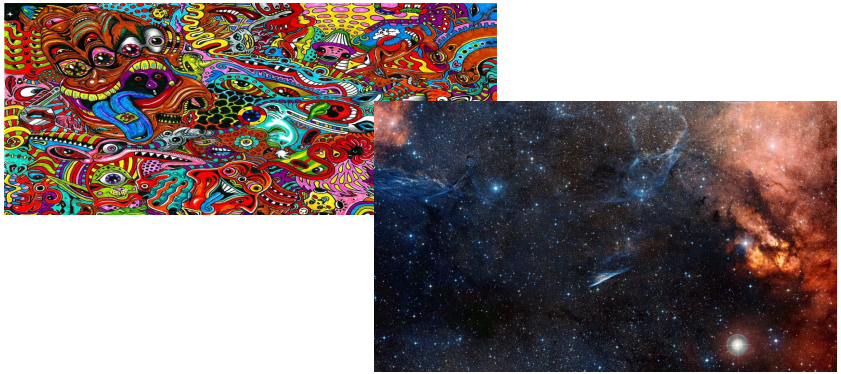
- 10 pixel width of black/white stripes
- 1024x600 dimensions (natively)
- Vertical Stripes



Sample Outputs of ImageStriper

Sample Output of ImageStriper with:

- 10 pixel width of stripes
- 1024x600 dimensions (natively)
- Vertical Stripes
- Using the two images below



Progress: Printer Calibration

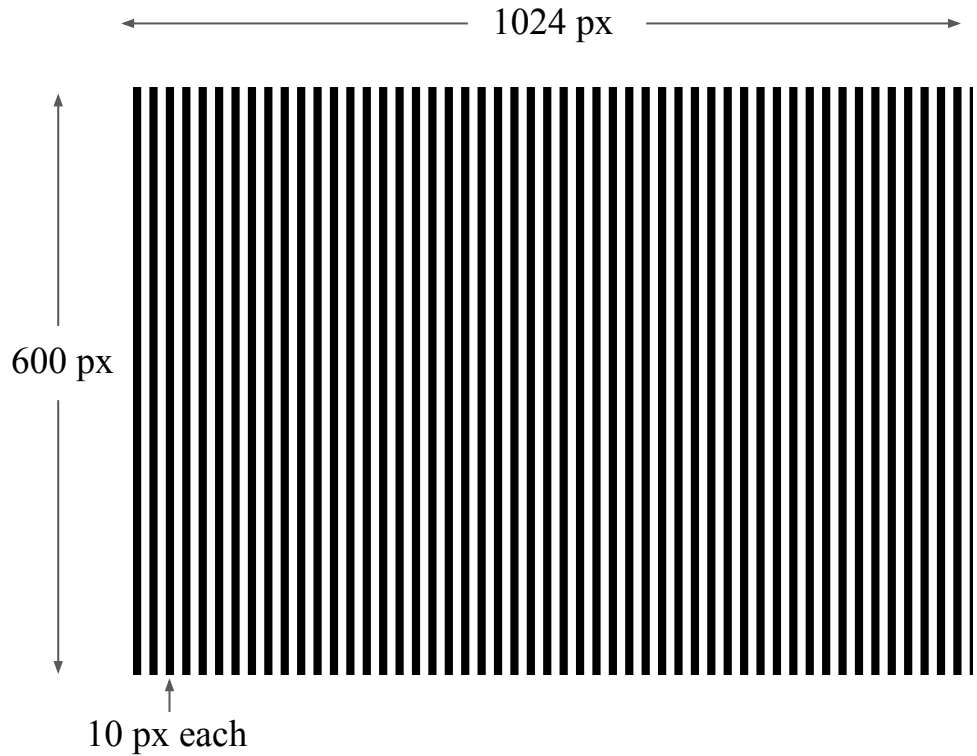
Parallax Barrier

Size: 1024 x 600 (px)

Resolution: 169.55 PPI

Stripe thickness: 10px

Display Scale: 100%

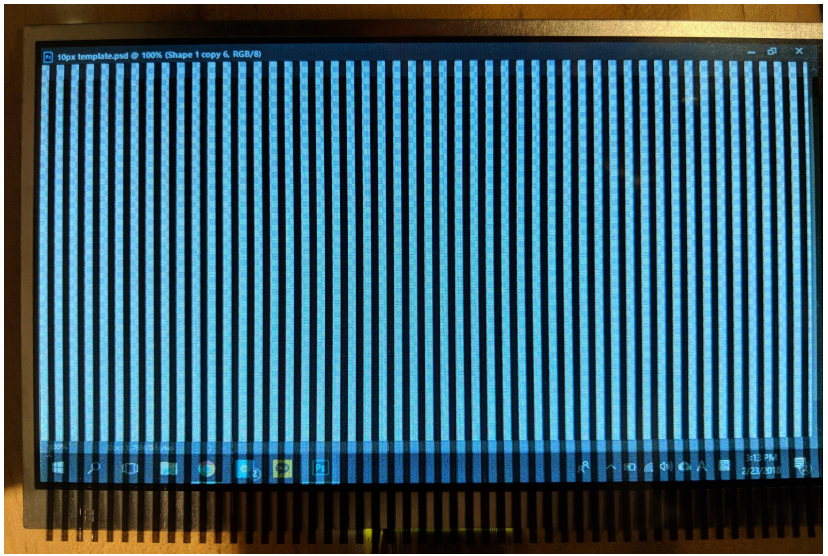


Progress: Printer Calibration

Calibration Method:

Directly compare the printed template with the displayed template.

After several interpolations, we concluded that the two images best match each other when the printed template was scaled to 100.5%



Progress: Printer Calibration

Parallax Barrier

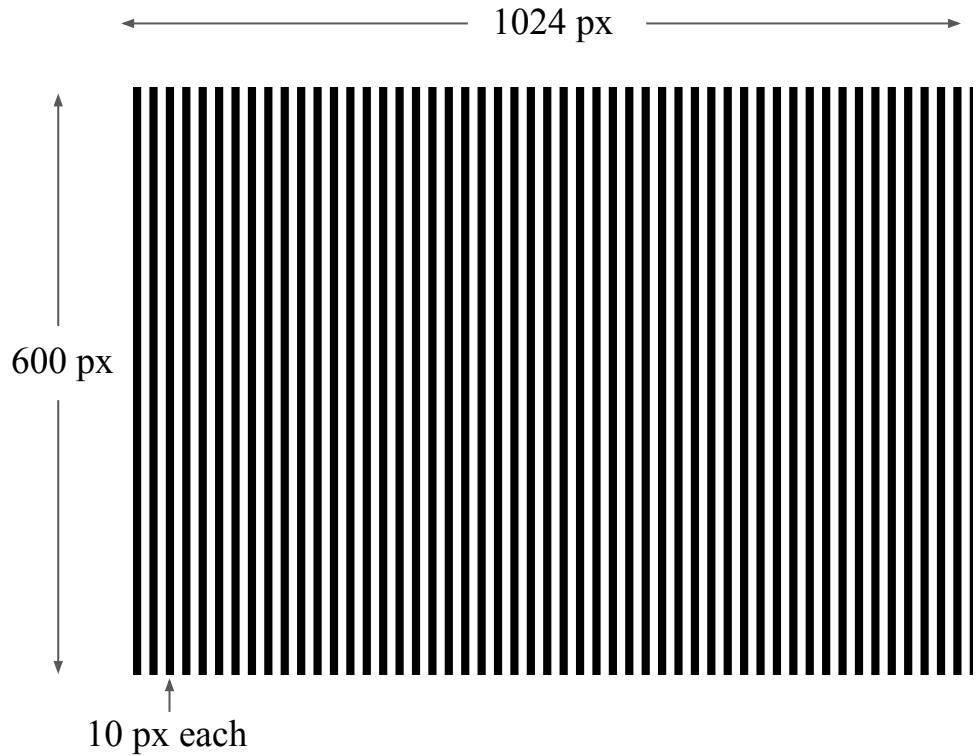
Size: 1024 x 600 (px)

Resolution: 169.55 PPI

Stripe thickness: 10px

Display Scale: 100%

Print Scale: 100.5%



Progress: Two-Color Experiment

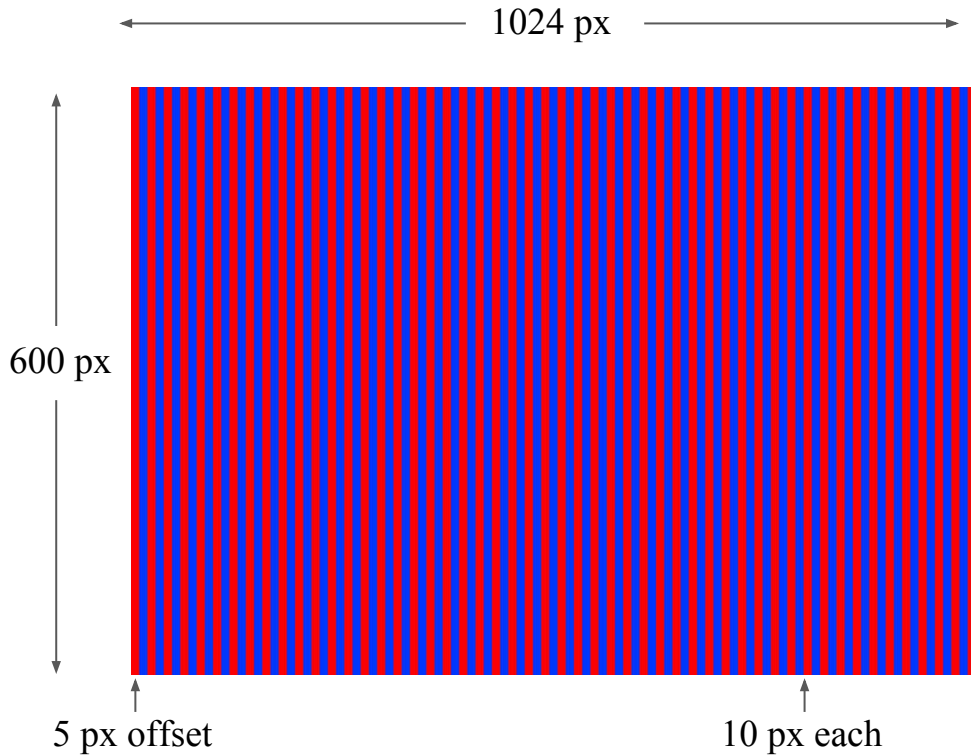
Two-Color Template

Size: 1024 x 600 (px)

Resolution: 169.55 PPI

Stripe thickness: 10px (first column is 5px)

Display Scale: 100%



Progress: Two-Color Experiment

Experiment Method:

Display a two-colored template and cover half of each stripe with a printed template.

When synced together, the stripes didn't match, the calibration must be incomplete.

However, we did get an effect very similar to what we expected as a result.

Progress: Printer Re-Calibration

Parallax Barrier

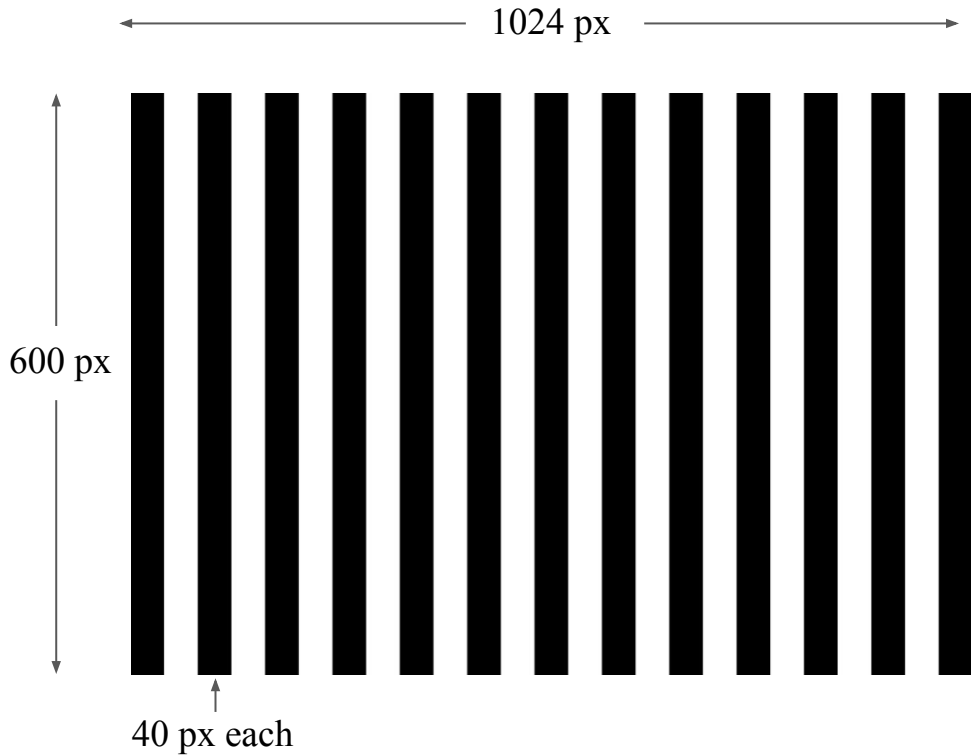
Size: 1024 x 600 (px)

Resolution: 169.55 PPI

Stripe thickness: 40px

Display Scale: 100%

Print Scale: 100%



Progress: Printer Re-Calibration

Calibration Method:

Thicker Parallax Barrier for easier measurement

Manually measure the thickness of the stripes and the space between the stripes

Compare measured thickness (see table) and computed thickness (0.5992.. cm)

Measurement for both stripe thickness and space to be 0.6 ± 0.01 cm.

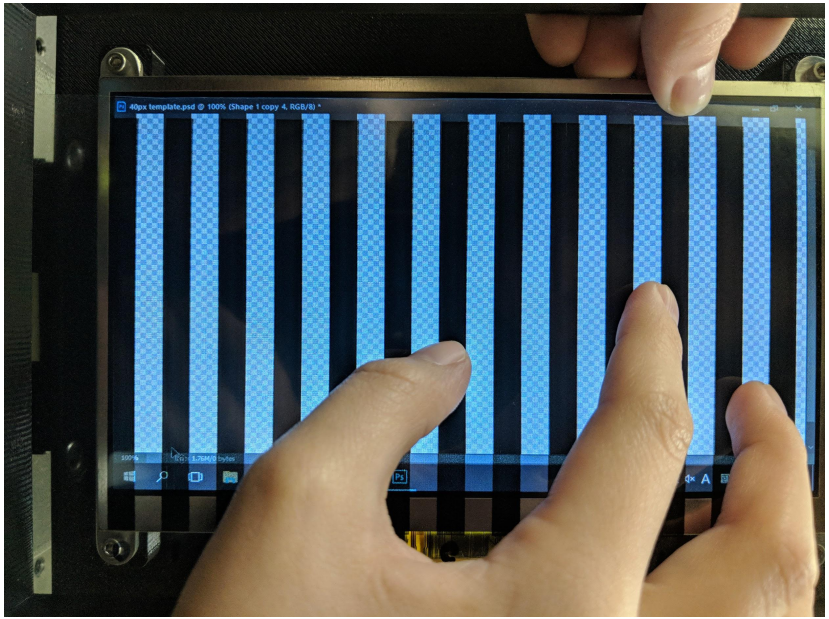
40px 100%		
no.	Stripe length (cm)	Spacing distance (cm)
1	0.6	0.59
2	0.59	0.61
3	0.6	0.6
4	0.6	0.6
5	0.6	0.6
6	0.6	0.6
7	0.6	0.6
8	0.6	0.6
9	0.6	0.6
10	0.6	0.6
11	0.6	0.6
12	0.6	0.6
13	0.6	0.6

Progress: Printer Re-Calibration

Result:

With calculation, the theoretical stripe thickness should be 0.5992 cm, which is 0.001% error.

However, the display and the printed parallax barrier did not match, which led to conclusion that the factor affecting the “matching” is the viewing angle rather than the printing quality.



Next Semester Goals

All of the necessary tool have been gathered:

- Hardware (printer, mount, display, raspberry pi)
- Software (creating images, calculating parameters)

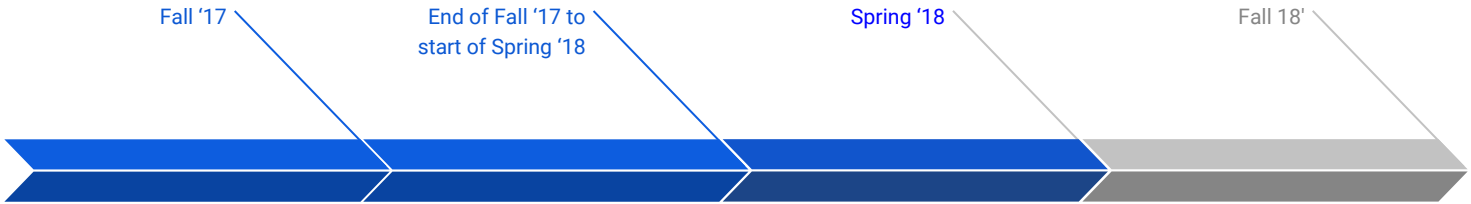
We are finally able to begin our series of experiments with different parameters

Define test variable, calculate dependent variables, create barrier, implement

Rinse and Repeat

Goal: create one barrier with correct behaviors to ensure feasibility of research

Conclusion


Fall ‘17
End of Fall ‘17 to start of Spring ‘18
Spring ‘18
Fall 18'
Initial Research/Understanding Concept
Obtain hardware and measure parameters
Familiarize with software
Testing

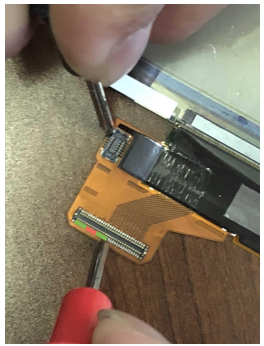
HoloJackets Sub-teams

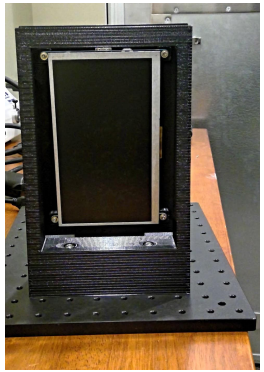
Images
Image Capture

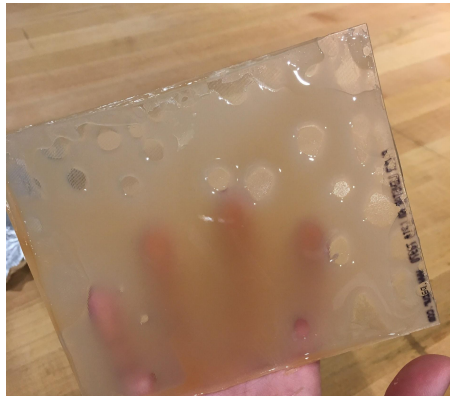
Electronics
High Resolution Display Group

Display
Parallax Barrier

Lens
Micro-Optics

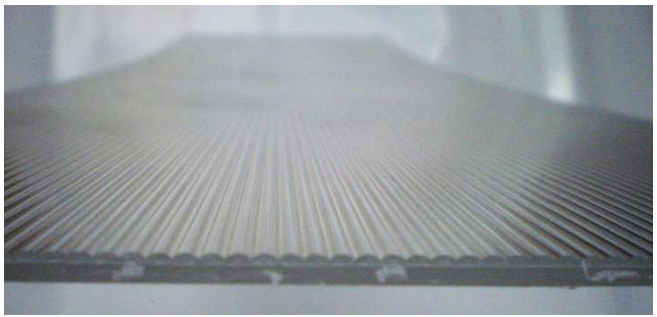


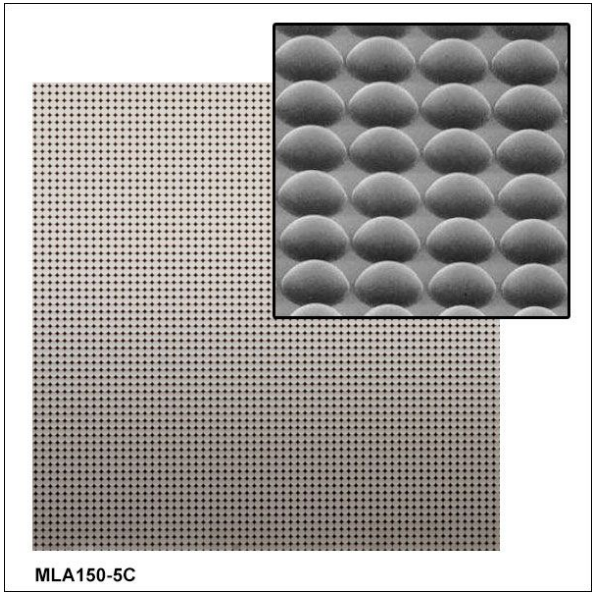




Progress: Micro-Optics

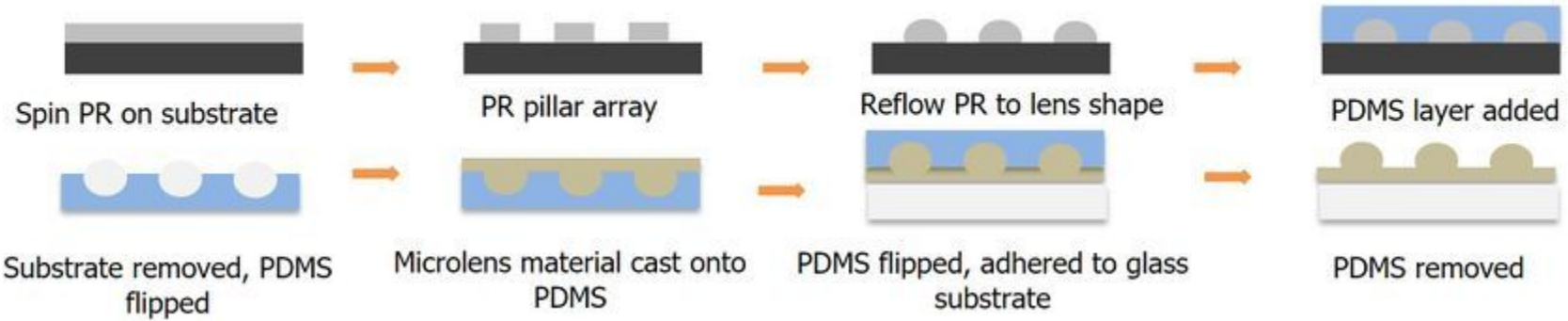
- Goal: Fabricate microlenses that will project a 3D display.
- Two types of lens fabrication:
  - Lenticular sheets (bottom left)
  - Microlens arrays (bottom right)





Micro-Optics

- Fabricate microlenses in the clean rooms using this method:



- Note: PR is shorthand for photoresist

Micro-Optics: Masters

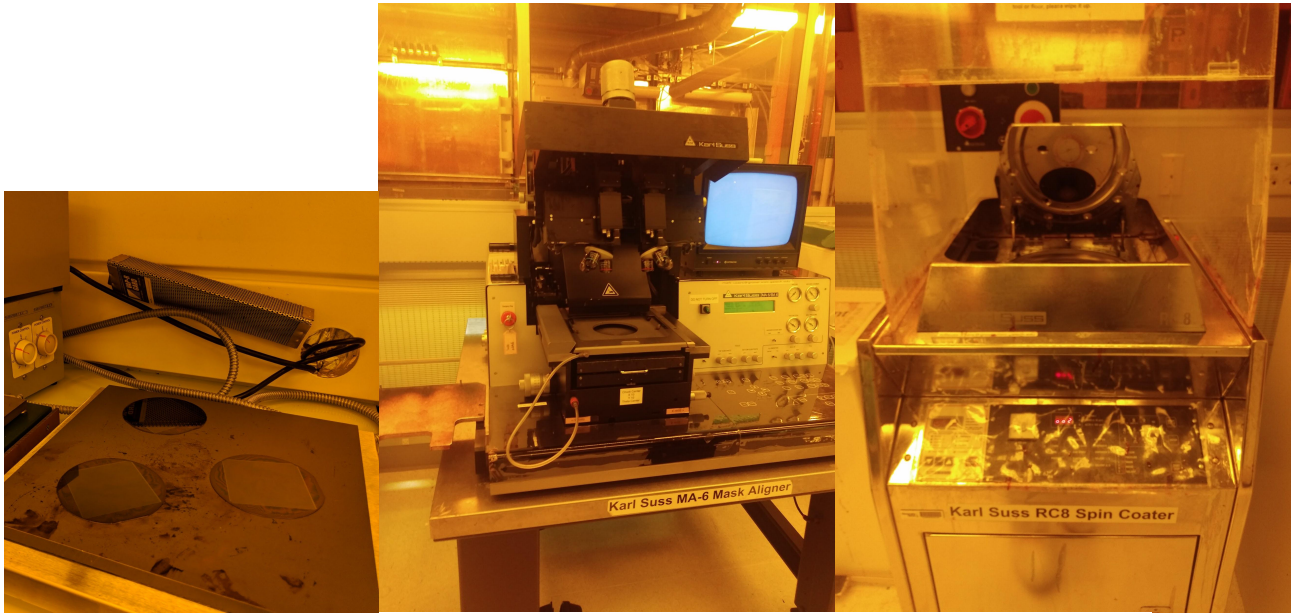
- Materials Used

Material	Polymer Type	Components	Cure Time	Viscosity
ReoFlex 30/40	Urethane	Two	16 Hours	Medium
Heavy Duty	Epoxy	Two	6 Hours	Low

- Notes
  - Urethane used for negative master -> flexible, not easily ripped
  - Epoxy used for positive master -> sturdy, clear
- Vacuum Chamber
  - Eliminates bubbles
  - Use cautiously for urethane, slightly cures it

Micro-Optics: Clean Room

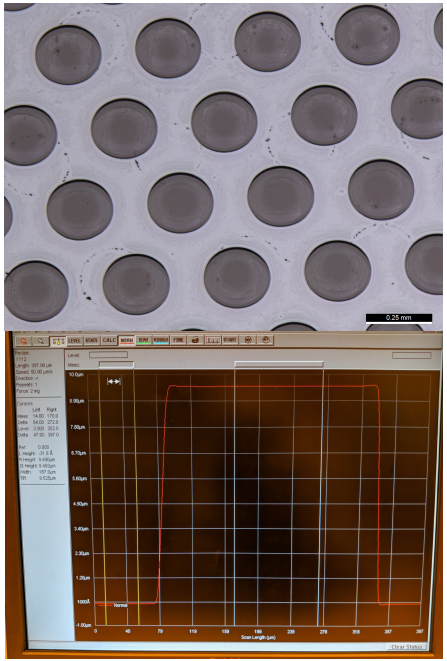
- Clean room training
  - Spin Coater
  - Mask Aligner
  - Profilometer
  - Hot Plate

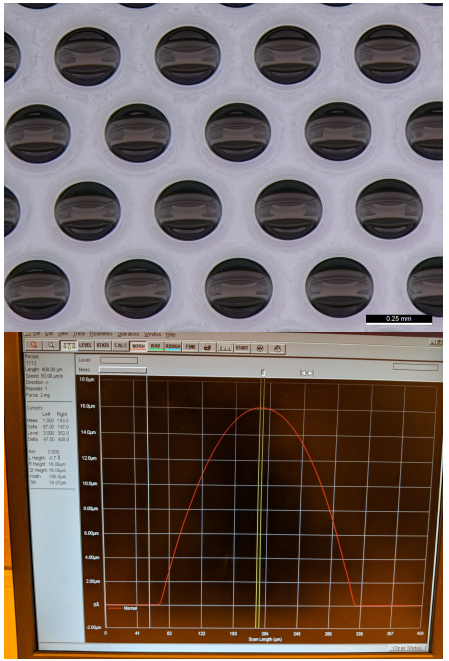


Hot Plate (left), mask aligner (middle), and spin coater (right)

Micro-Optics: Results

Original lenticular sheet (top) and epoxy master (bottom)





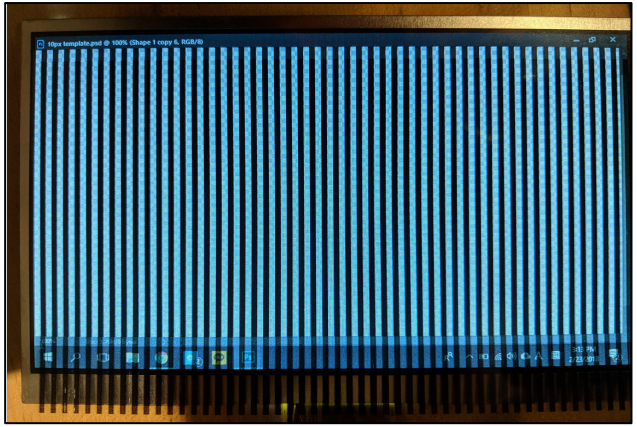
Microlenses before (top left) and after (top right) reflow. The pictures below are their respective profilometer results.

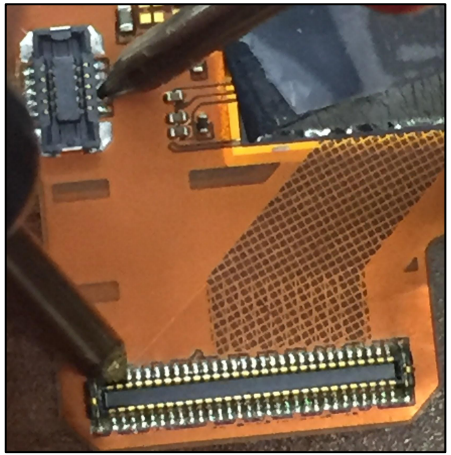
Looking Forward

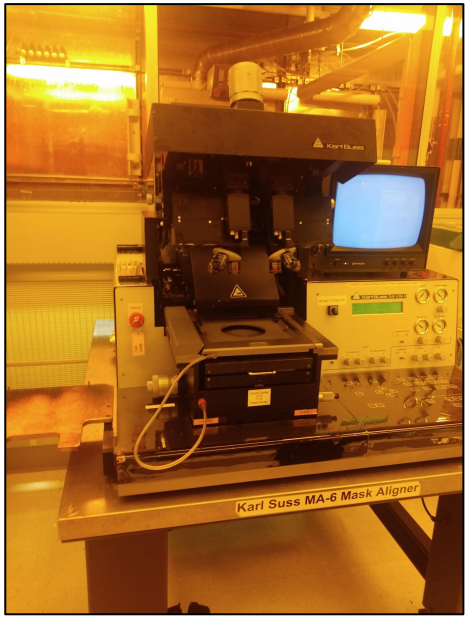
- Merging Clean Room and Masters
    - Masters have been made with lenticular sheets
    - Experiment with rectangular wafers
    - Now, we look to move toward making masters from the silicon wafers developed in the clean room

Summary

- Image Capture
- High Resolution Display Group
- Parallax Barrier
- Micro-Optics







Thank You

What does 3D mean?

There are four major depth cues that the human brain uses to gain true 3D sensation.

Some 3D display devices can provide all of these physical depth cues, while other autostereoscopic 3D display techniques may not be able to.

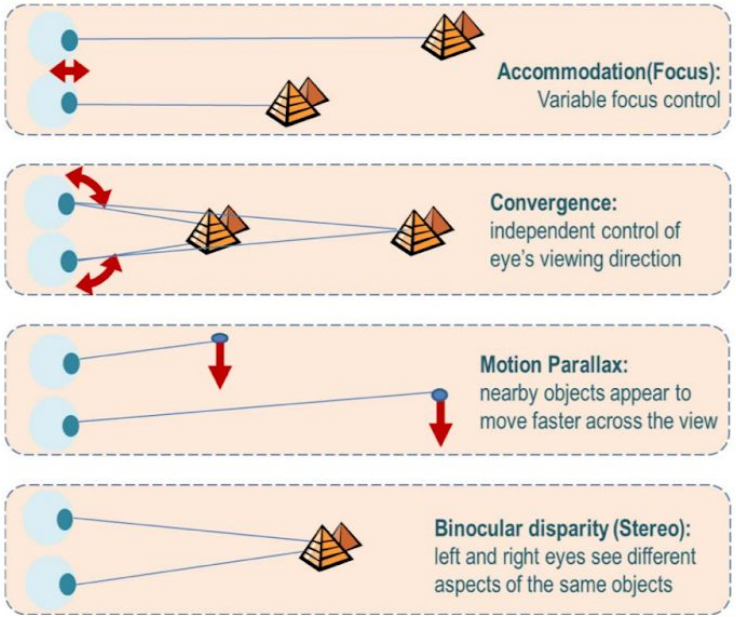


Illustration of four major physical depth cues.

What does 3D mean?

The human brain can also gain a 3D sensation by extracting psychological depth cues from 2D monocular images.

Different depth cues have different effects at different stand-off viewing distances.

It's often difficult for a 3D display device to provide all the physical and psychological depth cues simultaneously.

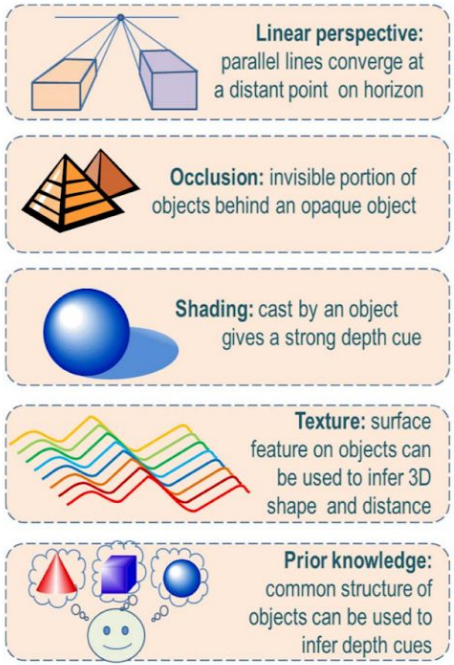


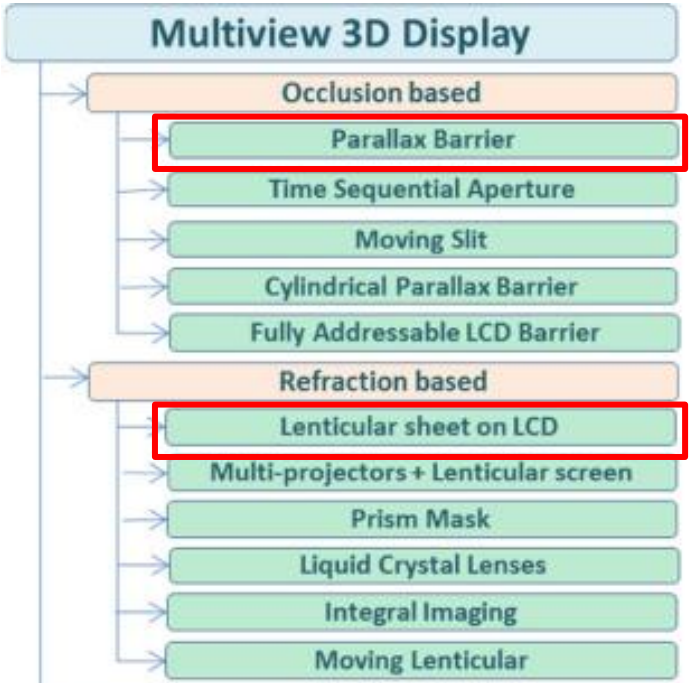
Illustration of psychological depth cues from 2D monocular images.

Implementation Strategies of Multiview 3D Displays

A multiview autostereoscopic 3D display system is able to produce different images in multiple angular positions.

No special eyewear is needed.

There are numerous implementation strategies, but our team is focused in Parallax Barrier and Lenticular sheet on LCD classes.



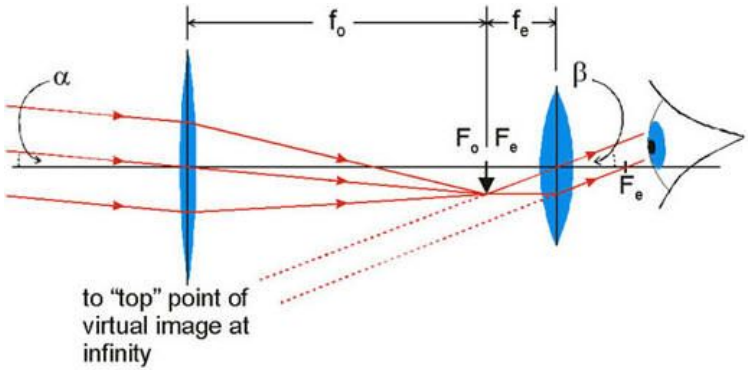
source

Optics Fundamentals: Ray Diagram

The line of sight principle suggests that in order to view an image of an object in a mirror, a person must sight along a line at the image of the object.

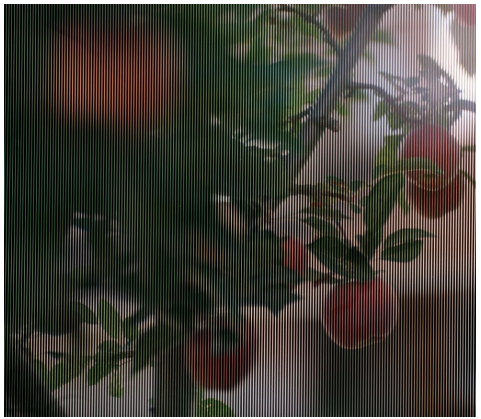
The light from the object reflects off the mirror according to the law of reflection and travels to the person's eye.

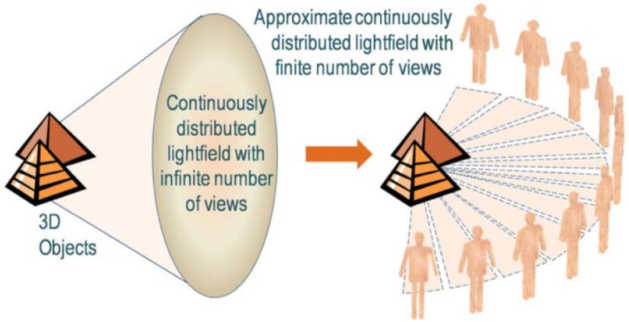
A ray diagram is a diagram that traces the path that light takes in order for a person to view a point on the image of an object.

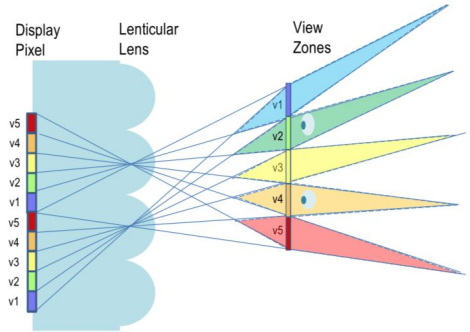


General Terminology

- Striped Images
- Multiview Displays
  - Occlusion-Based (parallax barrier)
  - Refraction-Based (lenticular)



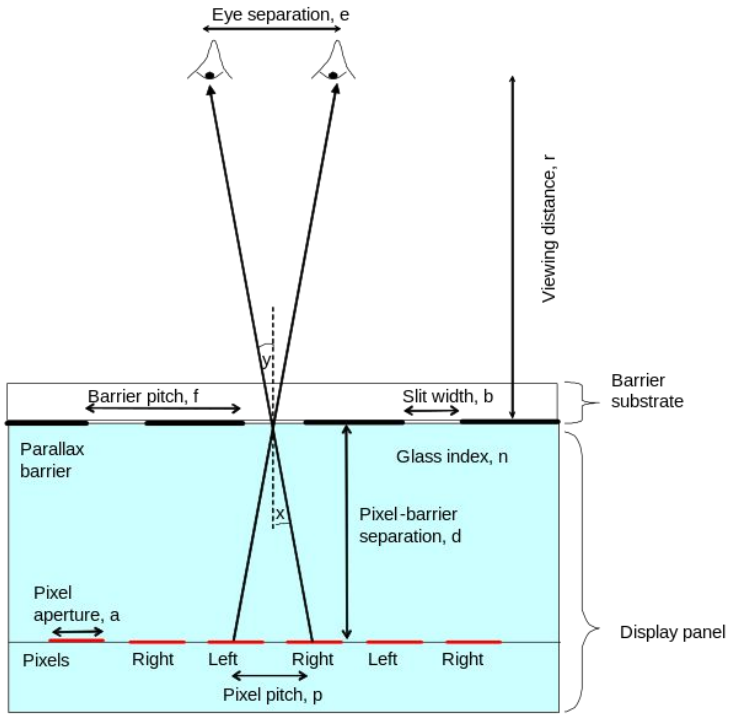




Optics Fundamentals: Parallax Barrier Diagram

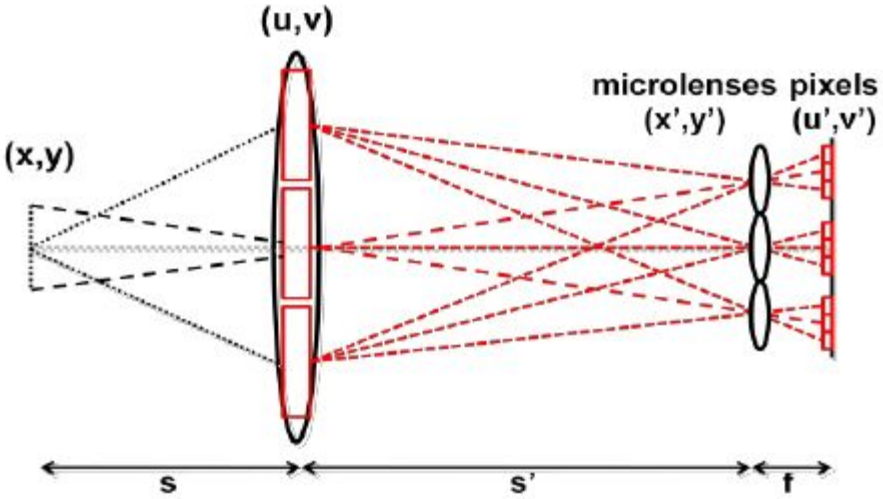
Pixel aperture, pixel pitch, and pixel-barrier separation is needed to find the barrier pitch and slit width.

Application of **solid angle** is required.

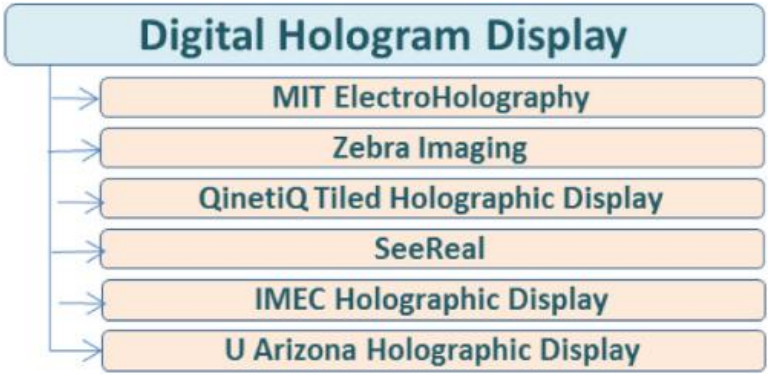


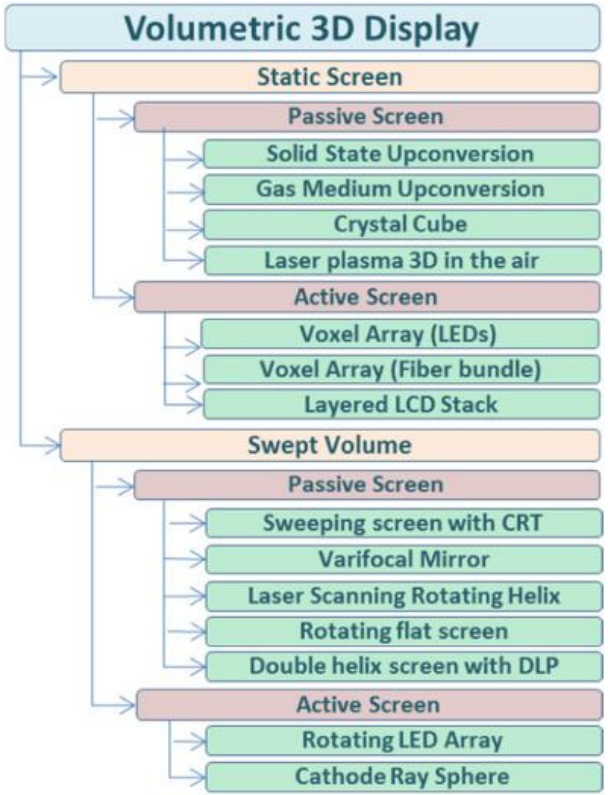
4-Dimensional Views

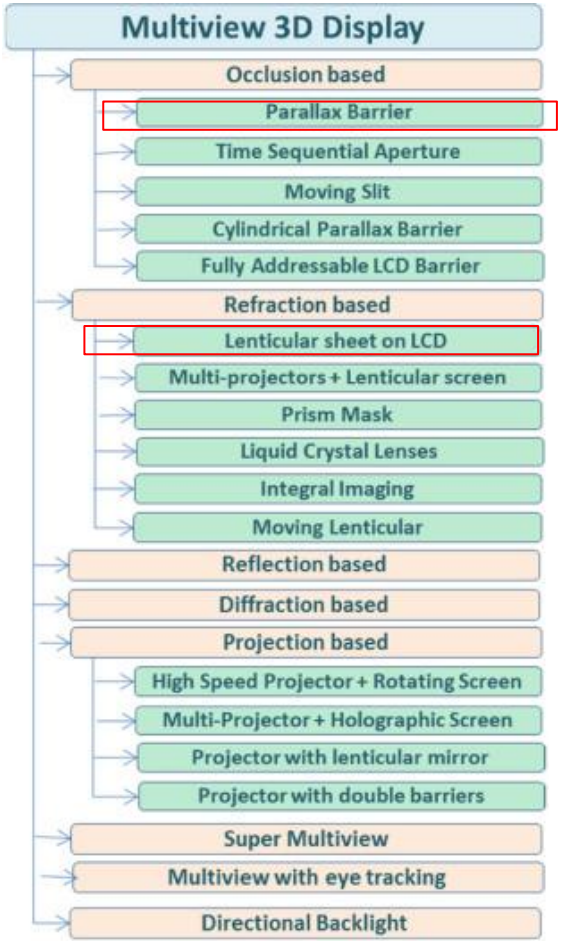
- Left Right
- Up down



Optics Fundamentals-Types of Displays




Volumetric 3D Display
Static Screen
Passive Screen
Solid State Upconversion
Gas Medium Upconversion
Crystal Cube
Laser plasma 3D in the air
Active Screen
Voxel Array (LEDs)
Voxel Array (Fiber bundle)
Layered LCD Stack
Swept Volume
Passive Screen
Sweeping screen with CRT
Varifocal Mirror
Laser Scanning Rotating Helix
Rotating flat screen
Double helix screen with DLP
Active Screen
Rotating LED Array
Cathode Ray Sphere


Multiview 3D Display
Occlusion based
Parallax Barrier
Time Sequential Aperture
Moving Slit
Cylindrical Parallax Barrier
Fully Addressable LCD Barrier
Refraction based
Lenticular sheet on LCD
Multi-projectors + Lenticular screen
Prism Mask
Liquid Crystal Lenses
Integral Imaging
Moving Lenticular
Reflection based
Diffraction based
Projection based
High Speed Projector + Rotating Screen
Multi-Projector + Holographic Screen
Projector with lenticular mirror
Projector with double barriers
Super Multiview
Multiview with eye tracking
Directional Backlight