

# BME 400 Fall 2022 iPhone Virtual Reality Training Model Complete Notebook

PDF Version generated by

Sam Neuman

on

Dec 14, 2022 @11:14 PM CST

## Table of Contents

Project Information	2
Team contact Information	2
Project description	3
Product Design Specification	4
Design Matrix	9
Previous Lab Archives	10
2022/09/15 - BME 300 FA21	10
2022/09/15 - BME 301 SP22	11
Team activities	12
Meeting Notes	12
2022/09/16 - Meeting Minutes	12
2022/09/09 - Meeting Minutes	14
2022/09/30 - Meeting Minutes	15
2022/10/05 - Meeting Minutes	16
2022/10/14 - Meeting Minutes	17
2022/10/17 - Meeting Minutes	18
2022/10/21 - Meeting Minutes	19
2022/11/14 - Meeting Minutes	20
2022/11/15 - Meeting Minutes	21
2022/11/18 - Meeting Minutes	22
2022/12/02 - Meeting Minutes	23
Design Process	25
2022/12/01 Two Phases of Progress	25
Materials and Expenses	26
2022/04/28 Materials Spreadsheet	26
2022/12/02 Materials Spreadsheet Fall 2022	27
Fabrication	28
2022/12/06 Fabrication Update	28
Testing and Results	29
Protocols	29
2022/12/07 Testing Protocol	29
2022/12/07 Testing Results	30
2022/09/23 - PDS	32
2022/09/30 - Design Matrix	37
Sam	38
Design Progress	38
2022/09/23 Mathematical Basis of Image Transposition	38
2022/09/29 The Single Frame View Design	39
2022/09/30 The Combined Frame View Preliminary Design	40
2022/11/02 MATLAB Source Code	42
2022/12/4 Mathematic Basis of Testing and Calibration of the Device	45
Research	48
2022/10/10 Reserach on Anaglyph Filters	48
2022/11/14 Research on VR Environment Coding	49
Haochen Wang	50
Design	50

2022/09/30 - Design ideas to combine hardware and software design .....	51
2022/10/05 - Python script .....	54
2022/11/17 - Unity with OpenCV .....	67
Research .....	75
2022/09/21 - Python image processing blog .....	75
2022/09/23-3D photography .....	78
2022/09/29-3D photography using context-aware layered depth inpainting .....	80
2022/10/17 - Application of Deliverable in Ophthalmology .....	82
Alex Vazquez .....	83
Design Ideas .....	83
2022/10/02- Potential Ideas .....	83
Research .....	84
2022/09/12- Background Project Research .....	84
2022/09/24-iPhone Design Research .....	86
2014/11/03-Template .....	87

**Team contact Information**

HAOCHEN WANG - Oct 12, 2022, 12:26 PM CDT

<b>Last Name</b>	<b>First Name</b>	<b>Role</b>	<b>E-mail</b>	<b>Phone</b>	<b>Office Room/Building</b>
Block	Walter	Advisor	wfblock@wisc.edu		
Shaffrey	Ellen	Client	shaffrey@wisc.edu		WIMR
Samuel	Poore	Client	poore@surgery.wisc.edu		WIMR
Neuman	Samuel	Leader & BWIG	ssneuman@wisc.edu		
Vazquez	Alexander	Communicator & BPAG	arvazquez@wisc.edu		
Wang	Haochen	BSAC & BWIG	haochen.wang@wisc.edu		
Albiter	Frida	BWIG	albiter@wisc.edu		



## Project description

---

HAOCHEN WANG - Oct 12, 2022, 12:28 PM CDT

**Course Number:** BME 400

**Project Name:** iPhone Virtual Reality Training Model for Microsurgical Practice

**Short Name:** iPhone VR

**Project description/problem statement:**

Due to high cost and importability of surgical microscopes, there is limited access to the microscopes and thus restricted training opportunities for trainees. And this limitation is more significant in less developed regions. This project aims to deliver a smartphone-based solution that displays the microsurgery station with depth perception, with minimal latency in video stream, with an affordable price and with high portability. Additionally, the design should not compromise the awareness of the surrounding environment when performing microsurgery training, and ideally the video can be simultaneously shared with instructors.

**About the client:**

Dr. Ellen Shaffrey and Dr. Samuel Poore are plastic surgeons at the Wisconsin Institute for Research and are both very passionate about the project of designing an alternative for microsurgical practice. They have worked with multiple teams to design this project and are currently supporting the fourth semester of the project.



# Product Design Specification

---

HAOCHEN WANG - Dec 14, 2022, 4:24 PM CST

**Course Number:** BME 400

**Project Name:** iPhone Virtual Reality Training Model for Microsurgical Practice

**Short Name:** iPhone VR

## Product Design Specification (PDS)

**Title:** iPhone Virtual Reality Training Model for Microsurgical Practice.

Sam Neuman, Alex Vazquez, Haochen Wang

September 23th, 2022

**Function:**

This training model will make microsurgical training less expensive and more accessible to a wide range of users. It eliminates the need for an expensive surgical microscope by replacing it with a smartphone. The prototype will utilize the zoom functionality of the smartphone for the surgeon to clearly see sutures and tissues up close. By using a smartphone, it is also possible to stream the training to Zoom or a similar platform so training can occur virtually. The design will minimize lag time between the recording phone and projecting device for simultaneous view of both the trainee and observers, while increasing spatial awareness and depth perception via binocular live video.

**Client requirements:**

- Must allow for depth perception with regard to where the trainee's hands are in relation to the work site.
- Must create an image with high enough zoom and resolution to see sutures (0.070 mm in diameter) clearly [1]
- Must remain inexpensive so it is widely accessible to training surgeons
- Must produce a streaming resolution of at least 10.2 megapixels
- Must have a frame rate of at least 24 frames per second
- Must have a stream delay of no more than 0.5 seconds
- Should utilize full magnification power of the smartphone

**Design requirements:**

### 1. Physical and Operational Characteristics

**a. Performance requirements:**

- i. The device must be able to provide a clear image of the subject in a clinical environment. The device must be able to handle daily use and must be able to handle a load of at least 400g, the weight of the heaviest available smartphones.

**b. Safety:**

- i. The device should be out of the way of the surgeon to prevent interference during practice and contact with hazardous material encountered during the simulated surgery. The device also needs to be able to be sterilized in an efficient manner before and after each use.

**c. Accuracy and Reliability:**

- i. The device should be able to consistently maintain a magnification of 2x and the displayed magnification should be accurate with repeated trials. The device should display an accurate and clear image of the surgery area with minimal latency.

**d. Life in Service:**

- i. The device should withstand continued use over the duration of the training process, the longest of which can last up to 12 hours. The device should be able to withstand this use everyday over its lifespan, as many different trainees may use the device.

**e. Shelf Life:**

- i. The device should be stored in normal interior conditions for an indefinite amount of time. This means that the device should not be made out of materials that degrade over a short period of time (6 months, the duration of one semester) in absence of normal use. With continued use, the team would expect the smartphone being the limiting factor for the whole design. Thus, the final deliverable should have at least one year of lifespan, which matches the lithium battery warranty provided by Apple. [2]

**f. Operating Environment:**

- i. The product will most likely be used in a domestic or indoor environment, so the device will not be exposed to extreme conditions.
- ii. 0-35 ° C operating temperature, - 20-45 ° C nonoperating temperature, 5-95% non-condensing, relative humidity (the specification of iPhone 8, and more restriction may be applied as other hardware is introduced to the final deliverable) [3]
- iii. The person who will use this will be the trainee, which is the person who is practicing surgery using the iPhone, and the trainer(s) who is/are watching the trainee on the headset.
- iv. Potential splash of food dye, blood, in vitro tissues, etc. [4]
- v. Components that are exposed to the operation station shall not be malfunctioned upon such splash
- vi. Potential scratches from the surgical equipment, such as tweezers or needles.
- vii. The final deliverable should at least endure accidental damage from the aforementioned scenarios, while maintaining the resolution to recognize the suture

**g. Ergonomics:**

- i. The product can involve delicate technology, such as smart phones and laptops, so the same restrictions of force that cause those devices not to be damaged or break apply here.
- ii. For the iPhone 8, do not submerge in water greater than 1 meter and for longer than 30 minutes. [3]

**h. Size:**

- i. Should be able to be set up in an indoor living space (i.e. 10 x 10 sqft, approximately 3 x 3 meters)

**i. Weight:**

- i. Optimum weight: < 10lbs (approximately 4.5 kg). Must be easily transportable

**j. Materials:**

- i. No restrictions on material mechanics
- ii. Cannot be toxic upon skin contact or inhalation
- iii. Shall have minimal degradation resistance, such as from sunlight

**k. Aesthetics, Appearance, and Finish:**

- i. The color of the product should be dull so that it doesn't distract from the microsurgical practice it is intended for. The shape and form should be adjustable so that each user/consumer can place it into alternate positions to get a better and more comfortable practice for themselves. The texture of the finish should be flat and soft in order for it to be comfortable for the user and in order for it to not be a distraction.
- ii. Should simulate the working condition of an operation room with microscopes
- iii. Must not interfere with the operation and training performance of the user

**2. Production Characteristics****a. Quantity:**

- i. Tens of Thousands of units will be needed so that this can replace all current expensive training mechanisms for microsurgical practice for medical residents.

**b. Target Product Cost:**

- i. The target cost of the product is undetermined thus far until clients discuss but it will need to allow for an iPhone, a stand, and any attachment that is necessary to put over the camera to replicate microsurgery practice as best as possible. There are existing products whose costs are at least \$100,000 [5] which is drastically greater than the target cost. The prototype is a cheap alternative for medical students to use for remote training, using materials that are commonly owned.

**3. Miscellaneous****c. Standards and Specifications:**

- i. ISO 10936-1:2017
  - Specifies the requirements for microscopes used during surgical procedures, so the team must adhere to these specifications when creating a design. However, since this prototype will be used for practice purposes, the requirements many not all apply [6]

- ii. Code of Federal Regulations Title 21, Volume 8, Sec. 882.4525 Microsurgical instrument [7]
  - The final deliverable will fall into the Class I medical device category, which is exempt from the premarket notification procedures 510(k)
- iii. Code of Federal Regulations Title 21, Volume 8, Sec. 878.4700 Surgical microscope and accessories [8]
  - The final deliverable, under definition of this section, will be a Class I device. However, since the recording device in this design will be a DC powered smartphone, no more actions shall be made upon this regulation

**d. Customer:**

- i. The customer would prefer the delay of relaying the image to the headset to be minimized for enhance practicing technique (less than 0.5 s)
- ii. The quality of the camera while zooming should be clear enough to clearly see the material being worked upon. 2x zoom using an iPhone 11 Pro was tested to be the most practical. The requirement is that the trainee is able to see the suture, which is 0.070 mm [1]
- iii. The camera should be able to show the depth of the workspace in order to help determine the distance between the instruments being utilized and the suture on the workbench. This may require the use of two lenses to allow for a binocular view
- iv. The device should be comfortable to wear for extended periods of time

**e. User-related concerns:**

- i. As this is a device used for practice, there will be no requirements for patient confidentiality.
- ii. Sterilization should not be an issue with regard to the camera setup. However, it may be practical to clean the headset with a wipe between uses.
- iii. The design should be able to receive accommodations for users with visual impairments.

**f. Competition:**

i. Augmented Reality (Mixed Reality)

The Microsoft Hololens is a very complex device which allows for similar types of practice. However, the Hololens is much less accessible and much more expensive. This will be an alternative that is possible to use from many different remote locations. Meanwhile, mixed reality provided by Hololens is rather redundant for the purpose of the clients. [9]

ii. Exoscopic Platforms

Zeiss, Olympus and Mitaka are well known medical device providers for exoscopes, featuring high definition images of the field with 8x to 30x magnifying capability. However, the price varies from 0.2 to 1.5 million dollars, resulting in limited access for trainees from less developed regions [5].

**Reference**

- [1] B. M. A. A; "The surgical suture," *Aesthetic surgery journal*, Apr-2019. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/30869751/>. [Accessed: 18-Sep-2022].
- [2] Apple, "iPhone battery and performance," *Apple Support*, 24-Sep-2021. [Online]. Available: <https://support.apple.com/en-us/HT208387>. [Accessed: 19-Sep-2022].



- [3] Apple, “iPhone 8 - Technical Specifications,” Official Apple Support, 25-Mar-2021. [Online]. Available: [https://support.apple.com/kb/sp767?locale=en\\_US](https://support.apple.com/kb/sp767?locale=en_US). [Accessed: 22-Sep-2022].
- [4] “Dr. Sam POORE and team featured ON BTN LiveBIG Wisconsin,” Department of Surgery, 13-Jul-2020. [Online]. Available: <https://www.surgery.wisc.edu/2020/07/13/dr-sam-poore-and-team-featured-on-btn-livebig-wisconsin/>. [Accessed: 20-Sep-2022].
- [5] D. J. Langer, T. G. White, M. Schulder, J. A. Boockvar, M. Labib, and M. T. Lawton, “Advances in INTRAOPERATIVE Optics: A brief review of Current Exoscope platforms,” *Operative Neurosurgery*, vol. 19, no. 1, pp. 84–93, 2019.
- [6] “ISO 10936-1:2017,” ISO, 26-Jul-2017. [Online]. Available: <https://www.iso.org/standard/69819.html>. [Accessed: 18-Sep-2022].
- [7] CFR - Code of Federal Regulations Title 21 Sec. 882.4525, vol. 8. 2020.
- [8] CFR - Code of Federal Regulations Title 21 Sec. 878.4700, vol. 8. 2020.
- [9] “Microsoft mixed reality – Healthcare,” Microsoft Mixed Reality – Healthcare. [Online]. Available: <https://www.microsoft.com/en-us/hololens/industry-healthcare>. [Accessed: 17-Sep-2022].



**Course Number: BME 400**

**Project Name: iPhone Virtual Reality Training Model for Microsurgical Practice**

**Short Name: iPhone VR**

Criteria	Weight	Combined Frame View		Single Frame View						
		Raw Score	Score	Raw Score	Score					
Effectiveness (Latency)	25	3/5	15	5/5	25					
Sensitivity (Depth Perception)	20	4/5	16	5/5	20					
Ease of Use	20	4/5	20	5/5	20					
Cost	15	3/5	9	2/5	6					
Compatibility	10	4/5	8	4/5	8					
Frame Rate	5	4/5	4	2/5	2					
Durability	5	4/5	4	5/5	5					
<b>Total</b>	100	26/35	76	28/35	86					

**Notes on the Criteria**

- 1. Effectiveness(Time Lag):** this is given the highest weight, because the primary goal for this semester is to reduce the time delay in streaming the video to the user. Since both design would use a monitor with wired connection to the smartphone, the limiting factor for reducing the time delay will be the running time complexity. Since the Combine Frame View design requires an extra step for combining the images, it has a lower score than Single Frame View design.
- 2. Sensitivity (Depth Perception):** This is given the second highest weigh due to the importance of depth persepction. This gives the trainee the requisite knowledge of the location of their hands/tools during the mock procedure. Any inconsistency or misinput given to the user could cause issues in the training sequence that could translate to an actual procedure. When combining the individual views into one frame, Combined Frame View design may result in interference in the RGB signal of the object, which can then undermine the depth perception. Because the microsurgery practice is performed at a millimeter level of precision, we deduce that the interference can significantly impact the depth perception of the design. Therefore, Combined Frame View receives a lower score for the criteria.
- 3. Ease of Use:** Ease of use is determined by the time to implement our deisgn to the microsurgery practice prosedure. Since the Single Frame View design requiries adjustment on the program to match the frame rate of the monitor for display, it is given a lower score for the criterion.
- 4. Cost:** cost includes the cost for mimium rerquired processing power of the smartphone for video capture, the laptop for processing the video, and possibly the cost for the display monitor that is compatible with the design. Because the team expect relative high proessing speed for video processing, neither of the design receives a perfect score. However, because the single frame design requiries at least 2x fps than the Combined Frame View design to reach equivalent viewing results,
- 5. Compatibility:** This criteria gauges the designs ability to be used with current training equipment and any other hardware components involved
- 6. Frame Rate:** This is a quantitative measurement of the Frames per second that is output to the monitor. A higher number indicates a faster\* processing of the image and results in a better user experience.
- 7. Durability:** Refers to the ability of the design to withstand any wear/tear of use during normal operation.



# 2022/09/15 - BME 300 FA21

HAOCHEN WANG - Sep 15, 2022, 12:31 PM CDT

## BME Design-Fall 2021 - Nicholas Jacobson Complete Notebook

PDF Version generated by  
HAOCHEN WANG  
on  
Sep 15, 2022 @ 11:31 AM CDT

### Table of Contents

Project Information	2
Team/Project Information	2
Project Description	3
Team/Students	4
Client Meetings	4
2021/09/17 - First Client Meeting	4
2021/09/22 - Second Client Meeting (Lab Tour)	7
2021/10/01 - First Client Meeting	9
2021/10/08 - Second Client Meeting	10
2021/10/22 - 3rd Client Meeting	11
2021/11/12 - 4th Client Meeting	12
Advisor Meetings	13
2021/09/18 - Advisor Meeting	13
2021/09/17 - Second Advisor Meeting	14
2021/10/01 - Third Advisor Meeting	16
2021/10/28 - Fourth Advisor Meeting	16
2021/11/01 - 5th Advisor Meeting	17
2021/11/16 - Show and Tell and Advisor meeting	18
2021/11/21 - Sixth Advisor Meeting	19
Design Process	20
Materials and Expenses	21
Materials and Expenses	21
Fabricator	22
2021/11/18 - Testing on most feasible parameters	22
2021/10/28 - Fabrication Process	26
Fabrication Results	30
Protocols	30
2021/10/04 - Measurement on the belly	30
2021/10/04 - Modifications on the calculation and formula	32
Experimentation	33
2021/10/08 - Measurement on the belly	34
Project Files	37
Model/Equations	38
Research Notes	38
Biology and Physiology	38
2021/10/01 - Microsurgery in Rat Belly Population	38
2021/10/13 - Relationship between Rat belly using Microsurgery Techniques	39
Computing Design	40
2021/10/23 - Hand Motion Analysis Tool	40
2021/10/28 - Depth Perception with Single Camera	41
Microsurgery Training	42
2021/10/14 - Microsurgery Training Recommendations	42
Design Object	43
2021/10/01 - Rat Belly Heatmap Design	43
Training Documentation	44
Testing	45

[Download](#)

iPhone-VR\_BME-300\_Lab-Archive.pdf (88.6 MB) Lab Archive from BME 300 FA21



# 2022/09/15 - BME 301 SP22

HAOCHEN WANG - Sep 15, 2022, 12:32 PM CDT

## BME Design-Spring 2022 - HENRY PLAMONDON Complete Notebook

PDF Version generated by  
HAOCHEN WANG  
on  
Sep 15, 2022 @ 12:32 PM CDT

### Table of Contents

Project Information	2
Team/Project Information	2
Project Description	3
Team/Project Milestones	4
Client Meetings	4
2022/09/04 - First Client Meeting	4
2022/09/04 - Second Client Meeting	5
2022/09/05	6
2022/09/15 - Fourth Client Meeting	7
2022/09/29 - Fifth Client Meeting	8
Advisor Meetings	9
2022/09/04 - First Advisor Meeting	9
2022/09/11 - Second Advisor Meeting	10
2022/09/18 - Third Advisor Meeting	11
2022/09/04 - Fourth Advisor Meeting	12
2022/09/29 - Fifth Advisor Meeting	13
Design Process	14
2022/09/01 Drawing Material for Housing	14
2022/09/29 Brass and Teflon	15
Materials and Dimensions	16
478.0202 Material Specifications	16
Fabrication	17
Fabrication Results	18
Protocols	19
2022/09/08 Protocol for testing air depth perception	19
Experimentation	20
2022/09/08 Test with Prototype 1.2	20
2022/09/08 Testing with prototype 1 to determine the angle of interest	22
2022/09/08 Test with charts	26
Project Files	26
2022/09/11 Product Design Specifications	26
Henry Plamondon	26
Research Notes	28
Biology and Psychology	28
Computing Diagrams	28
Circuit Diagrams	28
2022/09/01 Specifications	28
Design Notes	27
Timing Documentation	28
Hardware Notes	28
Research Notes	28
Biology and Psychology	28
Computing Diagrams	40
Materials	41
2022/09/01 - Research on Gasp Pro Breath	41
2022/09/01 - Research on Tugh 1588	40

[Download](#)

iPhone-VR\_BME-301\_Lab-Archive.pdf (22.8 MB) Lab Archive from BME 301 SP22



## 2022/09/16 - Meeting Minutes

HAOCHEN WANG - Oct 12, 2022, 10:37 AM CDT

**Title:** Meeting notes

**Date:** 09/16/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen, Prof. Block

**Goals:** To go through progresses for the semester

**Content:**

Advisor meeting

1. Presentation: design choice to make AR
2. Decide within few weeks
3. Describe options that you have tried - talk to MS HoloLens 2
4. **Order the 3D goggles and talk about this in the client meeting**
  - a. Test out the goggles
5. Consider real-time imaging processing in python
  - a. openCV
  - b. See sample code below
6. Test out with students on campus
  - a. Possibly from another country
7. Test the learning time compared to the exoscopes
8. The end point is a paper about the effectiveness of the system and reduce training complexity
  - a. The paper will be due next semester
9. phone -> laptop communication -> MP4 realtime code (python) -> image stream clip to left and right with filter -> add to output (timing)
  - a. Latency (time from input to output)
    - i. Come up with a requirement(30 fps, 0.1 s delay max)
  - b. Reliability (how often the laptop will fall behind during tasks)
    - i. Catch up with the real time processing if there is a time delay
    - ii. Priority in the task queue
10. Find real time processing software
11. Filtering is parallelizable
  - a. Analysis on the computational power (e.g: GPU)
12. **Learning image processing**
  - a. From python
  - b. Add filters to the images
13. Recommended restaurant: Stadium Takeout, Kosharie's

Client meeting

1. Goal is to use iPhone as a microscope with depth perception

- a. Used to use VR set, but there was too much lag time
2. Last year worked on making an attachment on the smartphone to split the view
  - a. Refine the mirror angle to achieve better results
3. Develop a larger prototype
4. Distance of the cameras determine the depth perception
  - a. Further distance, better depth perception
  - b. 63 mm apart
  - c. C920 cameras with raspberry pi**
  - d. Oculus web server for computational work
5. New monitor, not google cardboard.
6. Goals: have images stitched, complete astamimoses (?), ultimate goal is for med students/residents to do procedures.



## 2022/09/09 - Meeting Minutes

---

HAOCHEN WANG - Oct 12, 2022, 10:36 AM CDT

**Title:** Meeting notes

**Date:** 09/09/22

**Content by:** Haochen Wang

**Present:** Sam, Haochen, Prof. Block

**Goals:** To setup expectations for this semester

**Content:**

Progress report and what to report to the advisor

- Team activities

- What I did what I figured out/work out

- Team leaders break up the work/significance/gaps

Story to tell for the project

- Train sb with the design versus the microscope, what was the impact

- Fill the gap in commercial field (pinpoint)



## 2022/09/30 - Meeting Minutes

---

HAOCHEN WANG - Oct 12, 2022, 10:40 AM CDT

**Title:** Meeting notes

**Date:** 09/30/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen, Prof. Block

**Goals:** To go through progresses for the semester

**Content:**

Advisor:

1. Augmented reality instead of VR (new specification)
2. Explain the two designs (merge/anaglyph filter/enhance depth perception with the python script)
  - a. We don't need high quality image because we don't operate on patients
  - b. Acquisition 2 pts, combine the channel, separating the channel
  - c. Constraints: just need to watch the screen during movie, but we need to augment
  - d. Time capacity when viewing the image to combine
    - i. Guy in Italy miss one eye
3. How 3D work
4. Brain adding two images to create depth perception
5. Keep the channels separate
6. Latency and frame rate (define numerically) go in to the design matrix
7. What's the display like, what are the requirements
8. More layer, better depth distinction
  - a. How RGB-D works

Client:

1. Consider the focus
2. Client used raspberry pi
3. Try magnification and check the processing time when zooming in
4. The time delay is still significant
  - a. Reduce the number of connections to further reduce the latency (ie.: do not connect to an extra monitor for display)
5. The goal is to have the deliverable ready in Feb 2023 before the clients go to South Africa
  - a. One unit is enough
6. Availability
  - a. Lab is available after 5 pm





## 2022/10/05 - Meeting Minutes

---

HAOCHEN WANG - Oct 12, 2022, 10:49 AM CDT

**Title:** Meeting notes

**Date:** 10/05/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen, Prof. Block

**Goals:** To have a lab tour and have hand-on experience with the microscope

**Content:**

Client meeting:

1. Had a tour in the client's lab
2. Had experience viewing the microsurgery station from microscopes, to set up expectations on the intended image quality and magnification from the design



## 2022/10/14 - Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:16 PM CST

**Title:** Meeting notes

**Date:** 10/14/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

Advisor

1. Train more people per unit time with fixed budget
  - a. How they can utilize the deliverable
  - b. Hand-visual coordination**
    - i. Elaborate this in the documentation
    - ii. Why you are developing this
2. Present two channels separately to each eye
  - a. Use iMovie etc. to create the sample video and try the anaglyph effect
3. Cheap viewer that displays both projected view and the real world at the same time
4. Add annotation on the actual printed prototype
5. Have sanity check on the current design (Option 1)
  - a. May still need the hololens 2
  - b. Make a VR lens (Option 2)
- 6. Write up a page explaining the design with demo to send to person at Case Western**

Client

1. Demonstrated the current prototype with python



## 2022/10/17 - Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:16 PM CST

**Title:** Meeting notes

**Date:** 10/17/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

Client meeting

1. Using software/bluetooth for VR headset may cause significant delay
  - a. Lighting cord seems improve the delay
  - b. Bottleneck comes from the server delay
2. The mirror array focus may be not onto the correct point
  - a. Minimize the variables
3. Decide to meet two weeks later
  - a. Make the CAD housing
  - b. Consolidate the designs



## 2022/10/21 - Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:22 PM CST

**Title:** Meeting notes

**Date:** 10/21/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

Advisor meeting:

1. Project directly to the VR headset
2. Do the flipping?
  - a. Have the angles in the right orientation
3. Divide team up
  - a. VR headset
  - b. Bandpass filter approach
    - i. LOCI lab
4. Eventually determined by the user



## 2022/11/14 - Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:17 PM CST

**Title:** Meeting notes

**Date:** 11/14/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

Client meeting

1. Always save as the format in unity, and keep the original version in case of bug (save the project as)
2. Android and iPhone software development
3. Get a protocol
  - a. Can just get the Unity display without pushing the VR in the developer mode



## 2022/11/15 - Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:18 PM CST

**Title:** Meeting notes

**Date:** 11/15/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

Advisor meeting

1. Run the python script and then move the camera to see if the video is moving smoothly rather than skipping off the frames
2. Can use the code directly with referencing the author, need to check copyright from the author's page



## 2022/11/18 - Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:19 PM CST

**Title:** Meeting notes

**Date:** 11/18/22

**Content by:** Haochen Wang

**Present:** Sam, Alex, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

Advisor meeting

1. Send identical images to the VR headset to checkout if the code is working
2. Test movies
3. Get a VR headset



## 2022/12/02- Meeting Minutes

---

HAOCHEN WANG - Dec 14, 2022, 4:21 PM CST

**Title:** Meeting notes

**Date:** 12/02/22

**Content by:** Haochen Wang

**Present:** Sam, Haochen

**Goals:** To show current progress and plan for the next step

**Content:**

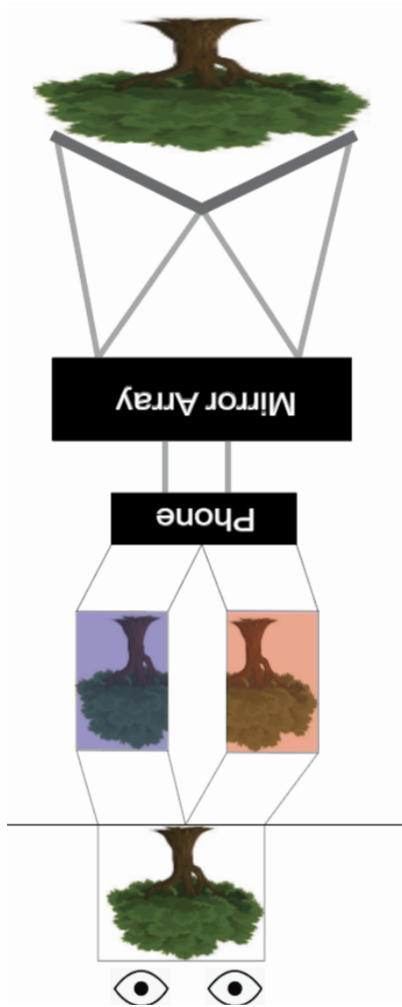
Client meeting

1. Testing
  - a. delay
  - b. Calibration position
2. Need to show unity on desktop
- 3.

Advisor meeting

1. Poster





2. 608-772-5642 sent a text
3. Show what stage we are having (image capture phase to streaming phase)
4. Take pictures for the poster
  - a. Unity platform, then write the code to sending off to each channel
  - b. Targeting to editing the image and shift the image as necessary
  - c. Provide a wide range of potential of imaging processing options during stream
  - d. Picture what we gonna do (1st figure, phases)
  - e. Second picture (left and right VR protocols) wifi connection, where we are right now (phone to unity - - > oculus)
  - f. Picture on a phone taken and the output from the unity interface
  - g. Do the real time demo at last
5. Outreach
  - a. Show children the vr experience
  - b. Safety, gaming (market for GPU and relatable deep learning)
  - c. 3D understanding of the surgery
  - d. No need to have a particular place
  - e. Potentially test with hands on experience with the design
  - f. Possibly making a new game?
  - g. reading small fonts/doing suture etc.
  - h. How does the tech enhance the task?
  - i. Augment the capability of human being
- 6.



## 2022/12/01 Two Phases of Progress

Sam Neuman - Dec 12, 2022, 9:58 AM CST

**Title:** The Two Phases of Progress

**Date:** 12/01/2022

**Content by:** Sam

**Present:** Sam, Haochen

**Goals:** Outline the two phases of the project

**Content:**

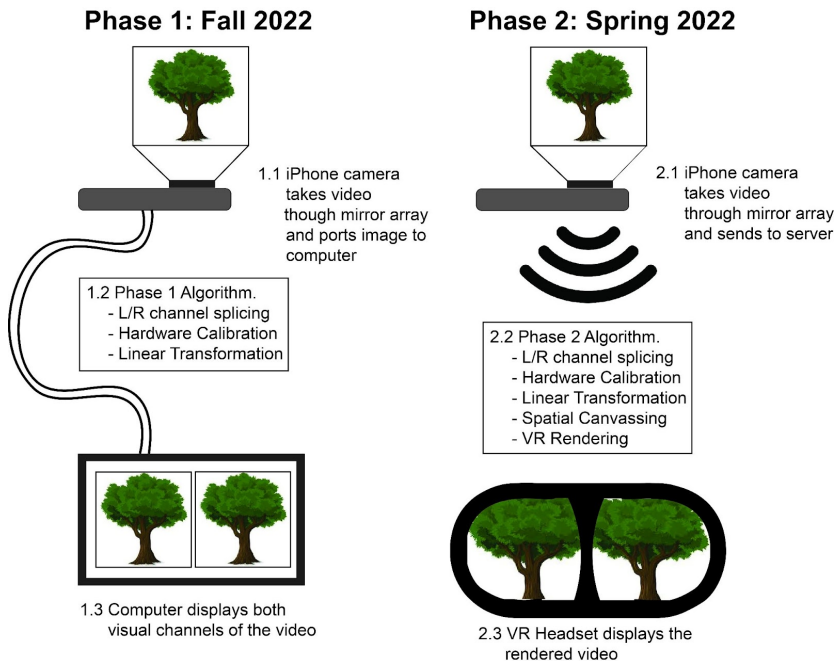


Figure 1. The two phases of the design's progress.

We propose a two-phase progress timeline for this project. Phase 1 involves proof-of-concept work. Specifically, we will design a system in which a smartphone can take an image of an object, port the image livestream to a computer via tether, and the computer performs modifications to the image, including channel splicing, hardware calibrations, and a linear transformation to the reflected channel. The computer then displays both hemispheres to the user in a way that is meaningful. Phase 2 is defined by the application of phase 1 to virtual reality. In phase 2, the smartphone will route the image through a local server to an Oculus VR headset. The headset take the role of the computer in performing its own image modifications. These modifications will be identical to those performed by the computer, with the addition of spatial canvassing and VR rendering.

### Conclusions/action items:

We propose to divide the project into two phases. Phase 1 will involve proof-of-concept work to develop a framework to transform the images into a meaningful picture to the user. Phase 2 will involve application of the work in phase 1 to the VR space. Hopefully, phase 2 will run smoothly. We anticipate most of the work in phase 2 will be troubleshooting VR programming.



## 2022/04/28 Materials Spreadsheet

---

EMMA KUPITZ - May 03, 2022, 10:20 PM CDT



[Download](#)

**Materials\_Spreadsheet\_-\_Sheet1.pdf (55.4 kB)**



## 2022/12/02 Materials Spreadsheet Fall 2022

Sam Neuman - Dec 12, 2022, 10:04 AM CST

**Title:** Material Spreadsheet for the Fall 2022 semester

**Date:** 12/02/2022

**Content by:** Sam

**Present:** Sam, Haochen

**Goals:** Outline the expenses accrued over the semester

**Content:**

Component	Use	Cost
3D Printed Material, Grey Pro ResinHousing		\$4
1x1 inch Mirrors	Reflect image	\$7.99
Oculus VR Headset	Stereoscopic display	Provided by team
Phone Boom Arm	Hold imaging device	Provided by client

Figure 1. Materials expenses for the fall 2022 semester.

We printed three prototypes of the mirror array out of PLA plastic. This totaled roughly \$4, and materials fees did not need to be paid. The mirrors used in the mirror array were reused from last year and were valued at about \$8. We also acquired an Oculus VR Headset for our own use that was used for testing of during phase 1; no expenses are associated with this. Finally, when considering the cost of the total apparatus, a phone arm may be useful, however we did not use one this semester.

### Conclusions/action items:

The final device will cost around \$300 for future customers, and the cost of the device is dominated by the purchase of a VR headset. There may be more affordable alternatives that could be considered, but our team found no success in locating an alternative. We hope this cost can be manageable if the design is ever brought to foreign countries for use. We are also satisfied with the relatively low cost of materials for assembly of the mirror apparatus, though more sturdy plastic may be used for printing the array assembly in the final market design.



## 2022/12/06 Fabrication Update

---

Sam Neuman - Dec 12, 2022, 11:38 AM CST

**Title:** Fabrication Update for the Fall 2022 Semester

**Date:** 12/06/2022

**Content by:** Sam

**Present:** Sam, Haochen, Alex

**Goals:** Give an update on fabrication progress over the semester

**Content:**

A total of three PLA prototypes of the mirror array were printed over the semester. These totaled less than \$10 to print, not including materials fees for the Makerspace. One of these prototypes were usable as the final design. Below is an image of the final prototype after fabrication.

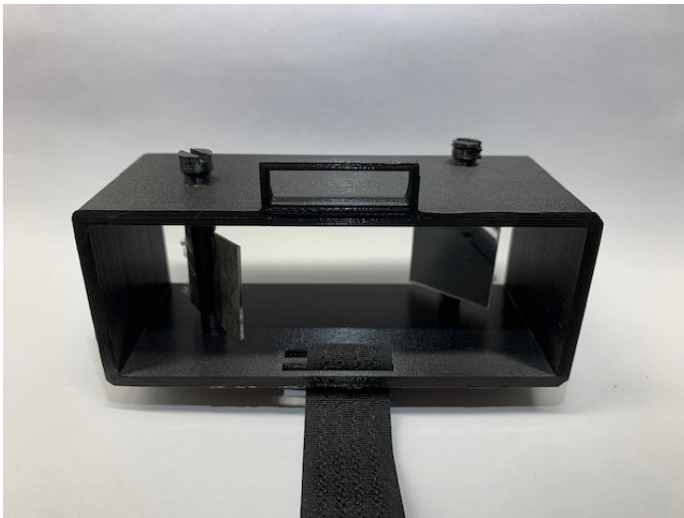


Figure 1. Wow. Just wow.

**Conclusions/action items:**

This semester's fabrication was rather uncomplicated. We printed a total of three prototypes, and because the final prototype met our expectations, we used it as our final prototype for the semester. PLA plastic was chosen because of its affordability and quickness of printing. We aim to use a higher quality plastic in the future, but PLA worked for this semester. We also used a velcro strap to secure the mirror array to a phone; no costs were associated with this addition.



## 2022/12/07 Testing Protocol

---

Sam Neuman - Dec 12, 2022, 11:55 AM CST

**Title:** Testing Protocol for Fall Semester

**Date:** 12/07/2022

**Content by:** Sam

**Present:** Sam, Haochen

**Goals:** Perform testing on the final prototype

**Content:**

We performed two experiments.

1. Quantification of the horizontal and vertical skew of the image. We drew a 1 cm x 1 cm square on graph paper and tuned the mirror array so that, at a focal distance of 25 cm, both non-reflected and reflected squares were in roughly the center of their corresponding hemispheres. We took three images. The images were loaded in ImageJ, and the disparity in distance between the inside of the squares and the vertical schism of the image as well as the disparity between the tops of the images were quantified using the draw function. We normalized the area between the non-reflected and reflected image by drawing an ROI around the non-reflected square and placed it in roughly the center of the reflected square. We compared data to a second square drawn ipsilateral to the non-reflected square. Statistics were done in MATLAB, and plots were generated in Prism.

2. We set up a live-stream through a local server to Unity canvas. The smartphone camera was focused on a stopwatch timer, and a second camera recorded both the stopwatch timer and the timer presented on Unity software. This experiment effectively quantified the effective time delay between the surgeon's hands and visual input. Multiple trials were run between five and thirty minutes.

**Conclusions/action items:**

We performed two experiments to quantify the success of our current design concept in meeting the client's requirements. Our first experiment quantified the horizontal and vertical skew of the reflected hemisphere relative to the non-reflected hemisphere. Our second experiment quantified the effective time delay between the surgeon's hands and their visual input. We also used this testing protocol to derive a set of equations that can be used to calibrate the horizontal and vertical mirror angles of the array such that automatic adjustments could be made in later stages of the device.



## 2022/12/07 Testing Results

Sam Neuman - Dec 14, 2022, 10:51 PM CST

**Title:** Results of Testing Fall 2022

**Date:** 12/07/2022

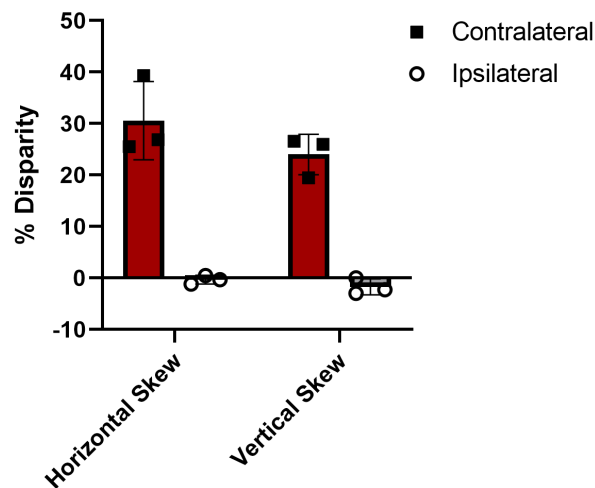
**Content by:** Sam

**Present:** Haochen, Sam

**Goals:** Record results of testing

**Content:**

Statistical significance was achieved in the degree of horizontal ( $p < 0.001$ ) and vertical ( $p < 0.001$ ) skew relative to an ipsilateral control object (Figure 13).



*Figure 1 Quantification of the Horizontal and Vertical Skew between reflected and non-reflected hemispheres relative to the non-reflected hemisphere. Relevant controls include second squares drawn at the same horizontal or vertical position as the experimental square to control for vertical or horizontal skew, respectively, as a result of image compression or imperfections in the mirror. (n=3,3)*

Testing on the algorithm with OpenCV and Python rendered a time delay of 0.23 +/- 0.01 seconds. This shows the relative time delay due to image processing. The delay of the video due to streaming was approximately 0.15 seconds -- this measurement is highly dependent on server capacity and network availability. The delay of the video equilibrated after about 10 seconds of use, and trials spanning >30 minutes showed consistent time delays with little variation. No lagging or frame-skipping was observed in any trials. An example cross-section of time delays over a 25 second stream is shown in Figure 14.

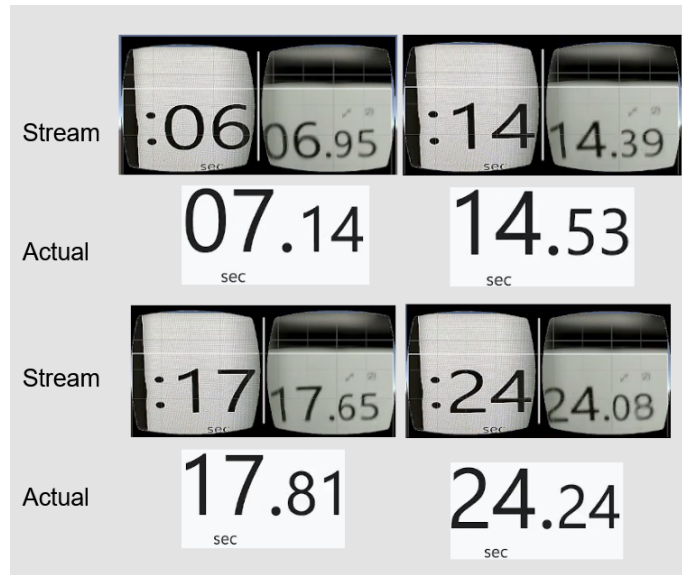


Figure 2. Time delay measurement for OpenCV on testing. The upper image shows separated left and right visual channels as they were displayed in Unity software, and the lower stopwatch shows the object being live-streamed.

**Conclusions/action items:**

We achieved statistical significant differences between vertical and horizontal skew between ipsilateral and contralateral hemispheres. This result was achieved despite normalization of the areas of the two testing squares between reflected and non-reflected hemispheres. These results will be crucial in informing the effectiveness of the device in its current iteration. These results are also useful for us in the development of normalization equations that can be used in the design of the mirror array and software to account for inaccuracies in image position and size.



**Title: Product Design Specification (PDS)****Date:** 09/23/2022**Content:**

## Product Design Specification (PDS)

**Title:** iPhone Virtual Reality Training Model for Microsurgical Practice.

Sam Neuman, Alex Vazquez, Haochen Wang

September 23th, 2022

**Function:**

This training model will make microsurgical training less expensive and more accessible to a wide range of users. It eliminates the need for an expensive surgical microscope by replacing it with a smartphone. The prototype will utilize the zoom functionality of the smartphone for the surgeon to clearly see sutures and tissues up close. By using a smartphone, it is also possible to stream the training to Zoom or a similar platform so training can occur virtually. The design will minimize lag time between the recording phone and projecting device for simultaneous view of both the trainee and observers, while increasing spatial awareness and depth perception via binocular live video.

**Client requirements:**

- Must allow for depth perception with regard to where the trainee's hands are in relation to the work site.
- Must create an image with high enough zoom and resolution to see sutures (0.070 mm in diameter) clearly [1]
- Must remain inexpensive so it is widely accessible to training surgeons
- Must produce a streaming resolution of at least 10.2 megapixels
- Must have a frame rate of at least 24 frames per second
- Must have a stream delay of no more than 0.5 seconds
- Should utilize full magnification power of the smartphone

**Design requirements:**

### 1. Physical and Operational Characteristics

**a. Performance requirements:**

- i. The device must be able to provide a clear image of the subject in a clinical environment. The device must be able to handle daily use and must be able to handle a load of at least 400g, the weight of the heaviest available smartphones.

**b. Safety:**

- i. The device should be out of the way of the surgeon to prevent interference during practice and contact with hazardous material encountered during the simulated surgery. The device also needs to be able to be sterilized in an efficient manner before and after each use.

**c. Accuracy and Reliability:**

- i. The device should be able to consistently maintain a magnification of 2x and the displayed magnification should be accurate with repeated trials. The device should display an accurate and clear image of the surgery area with minimal latency.

**d. Life in Service:**

- i. The device should withstand continued use over the duration of the training process, the longest of which can last up to 12 hours. The device should be able to withstand this use everyday over its lifespan, as many different trainees may use the device.

**e. Shelf Life:**

- i. The device should be stored in normal interior conditions for an indefinite amount of time. This means that the device should not be made out of materials that degrade over a short period of time (6 months, the duration of one semester) in absence of normal use. With continued use, the team would expect the smartphone being the limiting factor for the whole design. Thus, the final deliverable should have at least one year of lifespan, which matches the lithium battery warranty provided by Apple. [2]

**f. Operating Environment:**

- i. The product will most likely be used in a domestic or indoor environment, so the device will not be exposed to extreme conditions.
- ii. 0-35 ° C operating temperature, - 20-45 ° C nonoperating temperature, 5-95% non-condensing, relative humidity (the specification of iPhone 8, and more restriction may be applied as other hardware is introduced to the final deliverable) [3]
- iii. The person who will use this will be the trainee, which is the person who is practicing surgery using the iPhone, and the trainer(s) who is/are watching the trainee on the headset.
- iv. Potential splash of food dye, blood, in vitro tissues, etc. [4]
- v. Components that are exposed to the operation station shall not be malfunctioned upon such splash
- vi. Potential scratches from the surgical equipment, such as tweezers or needles.
- vii. The final deliverable should at least endure accidental damage from the aforementioned scenarios, while maintaining the resolution to recognize the suture

**g. Ergonomics:**

- i. The product can involve delicate technology, such as smart phones and laptops, so the same restrictions of force that cause those devices not to be damaged or break apply here.
- ii. For the iPhone 8, do not submerge in water greater than 1 meter and for longer than 30 minutes. [3]

**h. Size:**

- i. Should be able to be set up in an indoor living space (i.e. 10 x 10 sqft, approximately 3 x 3 meters)

**i. Weight:**

- i. Optimum weight: < 10lbs (approximately 4.5 kg). Must be easily transportable

**j. Materials:**

- i. No restrictions on material mechanics
- ii. Cannot be toxic upon skin contact or inhalation
- iii. Shall have minimal degradation resistance, such as from sunlight

**k. Aesthetics, Appearance, and Finish:**

- i. The color of the product should be dull so that it doesn't distract from the microsurgical practice it is intended for. The shape and form should be adjustable so that each user/consumer can place it into alternate positions to get a better and more comfortable practice for themselves. The texture of the finish should be flat and soft in order for it to be comfortable for the user and in order for it to not be a distraction.
- ii. Should simulate the working condition of an operation room with microscopes
- iii. Must not interfere with the operation and training performance of the user

**2. Production Characteristics****a. Quantity:**

- i. Tens of Thousands of units will be needed so that this can replace all current expensive training mechanisms for microsurgical practice for medical residents.

**b. Target Product Cost:**

- i. The target cost of the product is undetermined thus far until clients discuss but it will need to allow for an iPhone, a stand, and any attachment that is necessary to put over the camera to replicate microsurgery practice as best as possible. There are existing products whose costs are at least \$100,000 [5] which is drastically greater than the target cost. The prototype is a cheap alternative for medical students to use for remote training, using materials that are commonly owned.

**3. Miscellaneous****c. Standards and Specifications:**

- i. ISO 10936-1:2017

- Specifies the requirements for microscopes used during surgical procedures, so the team must adhere to these specifications when creating a design. However, since this prototype will be used for practice purposes, the requirements may not all apply [6]
- ii. Code of Federal Regulations Title 21, Volume 8, Sec. 882.4525 Microsurgical instrument [7]
  - The final deliverable will fall into the Class I medical device category, which is exempt from the premarket notification procedures 510(k)
- iii. Code of Federal Regulations Title 21, Volume 8, Sec. 878.4700 Surgical microscope and accessories [8]
  - The final deliverable, under definition of this section, will be a Class I device. However, since the recording device in this design will be a DC powered smartphone, no more actions shall be made upon this regulation

**d. Customer:**

- i. The customer would prefer the delay of relaying the image to the headset to be minimized for enhance practicing technique (less than 0.5 s)
- ii. The quality of the camera while zooming should be clear enough to clearly see the material being worked upon. 2x zoom using an iPhone 11 Pro was tested to be the most practical. The requirement is that the trainee is able to see the suture, which is 0.070 mm [1]
- iii. The camera should be able to show the depth of the workspace in order to help determine the distance between the instruments being utilized and the suture on the workbench. This may require the use of two lenses to allow for a binocular view
- iv. The device should be comfortable to wear for extended periods of time

**e. User-related concerns:**

- i. As this is a device used for practice, there will be no requirements for patient confidentiality.
- ii. Sterilization should not be an issue with regard to the camera setup. However, it may be practical to clean the headset with a wipe between uses.
- iii. The design should be able to receive accommodations for users with visual impairments.

**f. Competition:**

i. Augmented Reality (Mixed Reality)

The Microsoft HoloLens is a very complex device which allows for similar types of practice. However, the HoloLens is much less accessible and much more expensive. This will be an alternative that is possible to use from many different remote locations. Meanwhile, mixed reality provided by HoloLens is rather redundant for the purpose of the clients. [9]

ii. Exoscopic Platforms

Zeiss, Olympus and Mitaka are well known medical device providers for exoscopes, featuring high definition images of the field with 8x to 30x magnifying capability. However, the price varies from 0.2 to 1.5 million dollars, resulting in limited access for trainees from less developed regions [5].

**Reference**

- [1] B. M. A. A; "The surgical suture," *Aesthetic surgery journal*, Apr-2019. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/30869751/>. [Accessed: 18-Sep-2022].
- [2] Apple, "iPhone battery and performance," *Apple Support*, 24-Sep-2021. [Online]. Available: <https://support.apple.com/en-us/HT208387>. [Accessed: 19-Sep-2022].
- [3] Apple, "iPhone 8 - Technical Specifications," Official Apple Support, 25-Mar-2021. [Online]. Available: [https://support.apple.com/kb/sp767?locale=en\\_US](https://support.apple.com/kb/sp767?locale=en_US). [Accessed: 22-Sep-2022].
- [4] "Dr. Sam POORE and team featured ON BTN LiveBIG Wisconsin," Department of Surgery, 13-Jul-2020. [Online]. Available: <https://www.surgery.wisc.edu/2020/07/13/dr-sam-poore-and-team-featured-on-btn-livebig-wisconsin/>. [Accessed: 20-Sep-2022].
- [5] D. J. Langer, T. G. White, M. Schulder, J. A. Boockvar, M. Labib, and M. T. Lawton, "Advances in INTRAOPERATIVE Optics: A brief review of Current Exoscope platforms," *Operative Neurosurgery*, vol. 19, no. 1, pp. 84–93, 2019.
- [6] "ISO 10936-1:2017," ISO, 26-Jul-2017. [Online]. Available: <https://www.iso.org/standard/69819.html>. [Accessed: 18-Sep-2022].
- [7] CFR - Code of Federal Regulations Title 21 Sec. 882.4525, vol. 8. 2020.
- [8] CFR - Code of Federal Regulations Title 21 Sec. 878.4700, vol. 8. 2020.
- [9] "Microsoft mixed reality – Healthcare," Microsoft Mixed Reality – Healthcare. [Online]. Available: <https://www.microsoft.com/en-us/hololens/industry-healthcare>. [Accessed: 17-Sep-2022].



## 2022/09/30 - Design Matrix

HAOCHEN WANG - Oct 12, 2022, 12:21 PM CDT

### Title: Design Matrix for Methods to Merge Hardware and Software Design

Date: 09/30/22

Content:

Criteria	Weight	Combined Frame View		Single Frame View	
		Raw Score	Score	Raw Score	Score
<b>Effectiveness (Latency)</b>	25	3/5	15	5/5	25
<b>Sensitivity (Depth Perception)</b>	20	4/5	16	5/5	20
<b>Ease of Use</b>	20	4/5	20	5/5	20
<b>Cost</b>	15	3/5	9	2/5	6
<b>Compatibility</b>	10	4/5	8	4/5	8
<b>Frame Rate</b>	5	4/5	4	2/5	2
<b>Durability</b>	5	4/5	4	5/5	5
<b>Total</b>	100	26/35	76	28/35	86

**1. Effectiveness(Time Lag):** this is given the highest weight, because the primary goal for this semester is to reduce the time delay in streaming the video to the user. Since both design would use a monitor with wired connection to the smartphone, the limiting factor for reducing the time delay will be the running time complexity. Since the Combine Frame View design requires an extra step for combining the images, it has a lower score than Single Frame View design.

**2. Sensitivity (Depth Perception):** This is given the second highest weigh due to the importance of depth perseption. This gives the trainee the requisite knowledge of the location of their hands/tools during the mock procedure. Any inconsistency or misinput given to the user could cause issues in the training sequence that could translate to an actual procedure. When combining the individual views into one frame, Combined Frame View design may result in interference in the RGB signal of the object, which can then undermine the depth perception. Because the microsurgery practice is performed at a millimeter level of precision, we deduce that the interference can significantly impact the depth perception of the design. Therefore, Combined Frame View receives a lower score for the criteria.

**3. Ease of Use:** Ease of use is determined by the time to implement our deisgn to the microsurgery practice prosedure. Since the Single Frame View design requies adjustment on the program to match the frame rate of the monitor for display, it is given a lower score for the criterion.

**4. Cost:** cost includes the cost for mimium rerquired processing power of the smartphone for video capture, the laptop for processing the video, and possibly the cost for the display monitor that is compatible with the design. Because the team expect relative high proessing speed for video processing, neither of the design receives a perfect score. However, because the single frame design requies at least 2x fps than the Combined Frame View design to reach equivalent viewing results,

**5. Compatibility:** This criteria gauges the designs ability to be used with current training equipment and any other hardware components involved

**6. Frame Rate:** This is a quantitative measurement of the Frames per second that is output to the monitor. A higher number indicates a faster\* processing of the image and results in a better user experience.

**7. Durability:** Refers to the ability of the design to withstand any wear/tear of use during normal operation.



**Title:** Mathematical basis of image transposition for manipulating images from 2 input sources

**Date:** 23 September 2022

**Content by:** Sam

**Present:** n/a

**Goals:** Make a forward mathematical proof of the transformations of the images that would occur in the software.

**Content:**

## Mathematical Basis of Image Transposition for Surgical Practice Device

Samuel Neuman

Presented 23 September 2022

This report is intended to be the beginning of a discussion on the manipulations that would need to be made to two images taken from different angles and focused on a single object to combine them into one single image. This is neither a definitive discussion nor the final decision of the team's design direction.

*Proof.* We can represent two matrices that can be seen as modifications of the identity matrix. The number of each matrix is equal to the number of pixels in each of the two images. The first matrix gives the x and y coordinates of each pixel in an image:

$$\begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix} \quad (1)$$

The second matrix is a linear transformation that is meant to be applied to the position matrix:

$$\begin{bmatrix} \cos(\theta) & 0 \\ 0 & 1 \end{bmatrix} \quad (2)$$

Where theta is the angle of deflection of the second set of mirrors of the angle that the imaginary focal plane of the image makes with the plane orthogonal to the direction of the object being imaged.

Notice that when we multiply these two matrices, certain characteristics apply.

$$\begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} x\cos(\theta) & 0 \\ 0 & y \end{bmatrix} \quad (3)$$

Where

$$\det \begin{bmatrix} x\cos(\theta) & 0 \\ 0 & y \end{bmatrix} = xy \cos \theta = Area \quad (4)$$

Taking the determinate of matrix result of Eq.3, we see that this is always nonzero with reasonable parameters. This means that there is no nullity in any of the rank or column, meaning that there is no loss of dimension of resolution in the compression.  $\square$

Functionally, our computations will be limited by two factors. The first is the computing power of the machine doing these calculations; a mild compression of the pixel resolution may be needed to be able to produce truly real-time images that have no frustrating lag-time between reality and virtual reality. The second is a possible reduction in the computed frame rate. It is known that human eyes only perceive 16 frames/second; thus, if the user's cell phone camera captures images at a higher frame rate, it may be beneficial to have the program occasionally skip frames to save on computing power.

### Conclusions/action items:

In this entry, I propose the mathematical basis for manipulations that could be made to two images to combine them into one image. This derivation relies on the fact that the device used to take the pictures will involve a single camera and an apparatus of mirrors to split the image to a distance that is roughly the distance between human pupils. When implementing this math into image manipulations, it will also require matrices that carry the hexadecimal values of the color of each pixel. More, more manipulations will be required to average the euclidian distance (?) of the RGB profile of each pixel during compression, and the overlapping regions of the pixels in the "center" of the two images will likely need to be cropped and removed from the image entirely.



**Title:** Overview of the single frame view design

**Date:** 29 September 2022 (edited 11 October 2022)

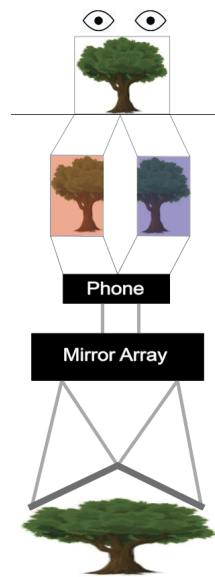
**Content by:** Sam

**Present:** Sam

**Goals:** Outline the design concept of the single frame view design.

**Content:**

The Single Frame View design presents the user with the left and right contributions of the final image simultaneously. Thus, this design is allowed to operate at our lower bound of frame rate of 60 fps. Hypothetically, the user's device may need to operate at a lower fps (I believe the human eye is only capable of resolving 16 fps - need to check this). To capture the three-dimensional form of the image being captured, three manipulations will need to be performed. First, the linear transformation will be applied to both the left and right images to compress them into normal-looking images. Next, the images will be cropped such that an equal area of each image is removed to create the final image; cropping will occur at an equal distance from the 'center' of the left and right images. Finally, an anaglyph filter will be applied to the images. This will create the perception of three dimensions in the images to allow for ascertainment of tissue features.



*Figure 1. The schematic of the single frame view. First, the camera acquires the images of the two inclined planes of the object. The phone first applies the linear transformation to flatten the image and crops out the overlapping regions of the two halves of the image. Finally, the phone applies an anaglyph filter (red on the leftmost shadows of the objects, blue on the rightmost shadows of the objects) to give the perception of three dimensions to the user. The left and right anaglyph contributions are overlaid onto a single image that is presented to the user.*

It is our estimate that the anaglyph filter will be a large drawback for the client. The implementation of 3D glasses will make some colors more difficult to see, like reds, blues, and greens. This would be less than useful in all contexts.

**Conclusions/action items:**

This entry aims to document the single frame view design of our preliminary designs. It involves taking a picture of the object through the mirror array before applying the linear transformation outlined in the previous entry, cropping the images, applying an anaglyph filter of red and blue to the left and right images respectively, and then stitching them to create the anaglyph shadow in the combined images. This single image is then presented to the user at optimally 60 fps. We anticipate that the requirements of 3D glasses for this design will be unattractive due to the loss of color resolution by the user as well as obstruction of peripheral vision by the glasses frames.





## 2022/09/30 The Combined Frame View Preliminary Design

Sam Neuman - Oct 11, 2022, 11:53 PM CDT

**Title:** The Combined Frame View Preliminary Design

**Date:** 30 September 2022 (edited 11 October 2022)

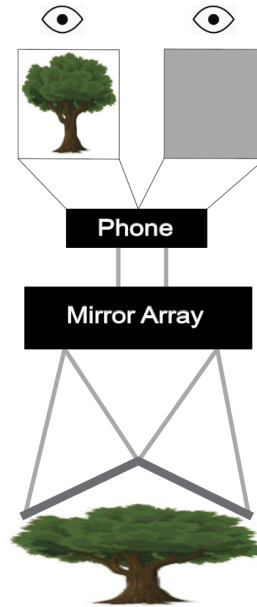
**Content by:** Sam

**Present:** Sam

**Goals:** Give an overview of the combined frame view preliminary design prior to the preliminary deliverables

**Content:**

The combined frame, like the single frame design, involves the simultaneous capture of both hemispheres through the mirror array. Note that the images taken in both hemispheres will have some degree of overlap (this is what helps them become 3D). The two hemispheres will be cropped such that each image contains the same landmarks and no distal parts of the images, but the two hemispheres will not be stitched together. Instead, they will be presented in an alternative fashion i. e. the left image will be presented quickly while the right image is not presented, then the right image will be presented without any semblance of the left image. Ideally, the images will be nearly identical with the exception of the lack of the linear transformation to flatten the images. The goal of this method is that, subliminally, the brain will fail to resolve differences between the images and will process each images as the contributions from each eye independently and then combine information from the two hemispheres to create the perception of three dimensions. In order for this to occur, a frame rate of 120 fps, or double the frame rate of the single view design, would need to be achieved.



*Figure 1. The combined view preliminary design. Both hemispheres of the images will be acquired simultaneously and cropped by the software to contain the same landmarks and thus be the same shape and encompass the same features. The phone screen will display only one of the left or right images to both eyes at one time in rapid succession, creating the illusion of each eye contributing its perspective to resolve a three dimensional image.*

Even more than the single frame view design, hardware capabilities will be a significant constraint on this design. Operating at a frame rate of 120 fps is optimistic, albeit in excess of the specifications of most computer monitors and phone screens. Otherwise, no accelerations of the computations will be needed, and this design concept is less computationally expensive due to the lack of the linear transformation of the images.

This design is based off of the 3D movies used in the early 2000s in Europe. People reported feeling nauseated by watching them. We are apprehensive towards this design for this reason - we believe that the alternative viewpoints, with a large angle theta between the image planes and the reference plane, will require the brain to amalgamate images with large disparities between them. Probably nauseating to be honest.

**Conclusions/action items:**

The combined frame view is a 'more extreme' version of the single frame view with the benefit of not having the same computational demands as the single frame view design. The combined frame view design, like the single frame view design, acquires a single image of both hemispheres through the mirror array. The phone does not apply the linear transformation, only cropping the images such that both images include the same

landmarks and are the same size as each other. The screen will then rapidly alternate between the left and right hemisphere images, creating the illusion of retinal disparity, allowing the user to subconsciously perceive three dimensions in the images. We expect that this design will be nauseating and unappealing to the clients for this reason as well as unrealistic to operate at 120 fps based on the hardware available to us.



## 2022/11/02 MATLAB Source Code

Sam Neuman - Dec 14, 2022, 11:06 PM CST

**Title:** MATLAB Source Code

**Date:** 2 November 2022 (accumulated over the semester)

**Content by:** Sam

**Present:** Sam and Haochen

**Goals:** Make an entry containing my proof-of-principle code developed in MATLAB

**Content:**

See below.

-----  
%% This is a MATLAB source code intended as a proof of principle of the image merging algorithm

```
clc; clear all; close all;
```

```
A = imread('IMG_20220930_110936491_HDR.jpg');
```

```
B = imread('IMG_20220930_110943041.jpg');
```

```
for i = 1 : 3
```

```
A = rot90(A);
```

```
B = rot90(B);
```

```
end
```

```
size_A = size(A);
```

```
size_B = size(B);
```

```
length_A = size_A(1,1,1);
```

```
width_A = size_A(1,2,1);
```

```
length_B = size_B(1,1,1);
```

```
width_B = size_B(1,2,1);
```

```
if (length_B > length_A)
```

```
B = B(1 : length_A , 1 : width_B);
```

```
else
```

```
A = A(1 : length_B , 1 : width_A);
```

```
end
```

```
size_A = size(A);
```

```
size_B = size(B);
```

```
length_A = size_A(1,1,1);
```

```
width_A = size_A(1,2,1);
```

```
length_B = size_B(1,1,1);
```

```
width_B = size_B(1,2,1);
```

```
if (width_B > width_A)
```

```
B = B(1 : length_B , 1 : width_A);
```

```
else
```

```
A = A(1 : length_A , 1 : width_B);
```

```
end
```

```

size_A = size(A);
size_B = size(B);
A = A(1 : size_A(1), 1 : size_A(2));
B = B(1 : size_B(1), 1 : size_B(2));

a = size(A);
b = size(B);
diff = 2200;

image_left = A(1 : a(1), 1 : (a(2) - diff));
image_left_center = A(1 : a(1), (a(2) - diff) : a(2));
image_right = B(1 : b(1), diff : b(2));
image_right_center = B(1 : b(1), 1 : diff);

C = imfuse(image_left_center, image_right_center, 'blend', 'Scaling', 'joint');
left_panel = (cat(2, image_left, C));

full_panel = cat(2, left_panel, image_right);
fig = figure(1);
imshow(full_panel);

print(fig, horzcat('Stitchpic', '.png'), '-dpng');

```

### %% Cosine Transformation

```

s=[2,3];
tform1 = maketform('affine',[s(1) 0 0; 0 s(2) 0; 0 0 1]);
Acos = imtransform(A, tform1);
Bcos = imtransform(B, tform1);

acos = size(Acos);
bcos = size(Bcos);
diffcos = 2500;

image_left = Acos(1 : acos(1), 1 : (acos(2) - diffcos));
image_left_center = Acos(1 : acos(1), (acos(2) - diffcos) : acos(2));
image_right = Bcos(1 : bcos(1), diffcos : bcos(2));
image_right_center = Bcos(1 : bcos(1), 1 : diffcos);

C = imfuse(image_left_center, image_right_center, 'blend', 'Scaling', 'joint');
left_panel = (cat(2, image_left, C));
imshow(left_panel);
full_panel = cat(2, left_panel, image_right);
fig2 = figure(2);
imshow(full_panel);

print(fig2, horzcat('Stitchpic_cosine', '.png'), '-dpng');

```

-----

**Conclusions/action items:**

This code was developed in MATLAB to perform proof-of-principle testing of our device. It acts to take two images of the same object from two different perspectives. It then splits the images into corresponding hemispheres and uses a grayscale average to blend the two images into a coherent picture. This code has been a living document over the course of the semester and has been useful to Haochen in integrating the image transformations into the working Python source code.

Sam Neuman - Dec 14, 2022, 11:07 PM CST

```
% This is a MATLAB source code intended as a proof of principle of the image
merging algorithm
clc; clear all; close all;
A = imread('D:\23022022\11061041_HDR.jpg');
B = imread('D:\23022022\11061041.jpg');
for i = 1 : 3
    A = rgb2gray(A);
    B = rgb2gray(B);
end
size_A = size(A);
size_B = size(B);
length_A = size_A(3,1);
width_A = size_A(2,1);
length_B = size_B(3,1);
width_B = size_B(2,1);
if (length_A > length_B)
    n = floor(length_A / length_B);
    A = A(1 : length_B * n, 1 : width_A);
end
size_A = size(A);
size_B = size(B);
length_A = size_A(3,1);
width_A = size_A(2,1);
length_B = size_B(3,1);
width_B = size_B(2,1);
if (width_B > width_A)
    n = floor(width_B / width_A);
    B = B(1 : length_B * n, 1 : width_B);
end
size_A = size(A);
size_B = size(B);
A = A(1 : size_A(1), 1 : size_A(2));
B = B(1 : size_B(1), 1 : size_B(2));
n = size(A);
m = size(B);
diff = 255;
image_left = A(1 : n(1), 1 : (n(2) - diff));
image_left_center = A(1 : n(1), (n(2) - diff) : n(2));
image_right = B(1 : m(1), diff : m(2));
image_right_center = B(1 : m(1), 1 : diff);
C = imfuse(image_left_center, image_right_center, 'blend', 'scaling', 'join');
left_panel = cat(2, image_left, C);
right_panel = cat(2, left_panel, image_right);
fig = figure();
imshow(right_panel);
```

[Download](#)

**BME400\_MatTest\_09302022.m (2.19 kB)**

Sam Neuman - Dec 14, 2022, 11:08 PM CST



[Download](#)

**Screenshot\_2022-12-14\_230821.png (83.4 kB)**



## 2022/12/4 Mathematic Basis of Testing and Calibration of the Device

---

Sam Neuman - Dec 12, 2022, 9:50 AM CST

**Title:** Discussion of the Mathematical Basis of Data Processing

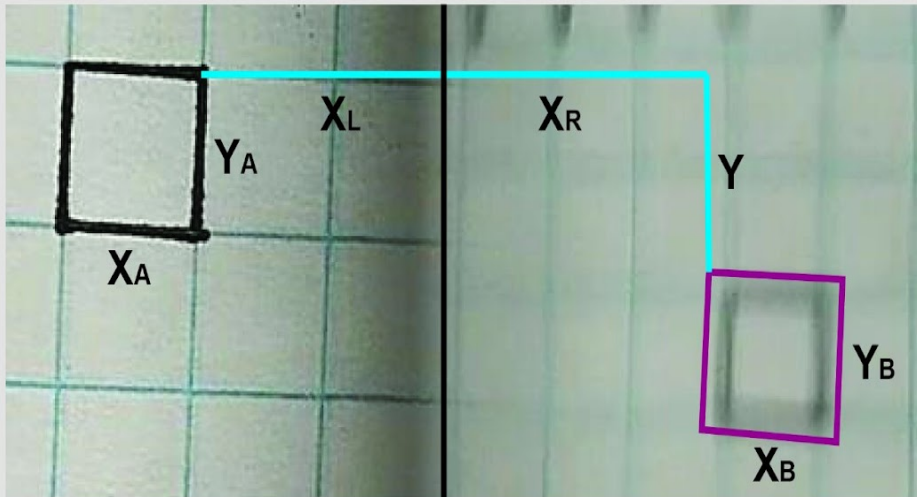
**Date:** 12/4/2022 (written 12/9/2022)

**Content by:** Sam

**Present:** Sam

**Goals:** Derive equations for angles of adjustment to the mirror array.

**Content:**



## Calibration Angles

$$\Delta\phi_x = F \times \arcsin(X_L - X_R)$$

$$\Delta\phi_y = F \times \arcsin(Y)$$

## Normalization

$$\mathbf{A} = \begin{bmatrix} X_A & 0 \\ 0 & Y_A \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} X_B & 0 \\ 0 & Y_B \end{bmatrix}$$

$$\text{define } \mathbf{S} \in \mathbb{R} = \begin{bmatrix} \cos(\theta) & 0 \\ 0 & 1 \end{bmatrix}$$

$$\det(\mathbf{A}) = \det(\mathbf{B} \times \mathbf{S}) = \text{Area} = xy\cos(\theta)$$

Figure 1: Top. An example image taken during testing of the device. A single 1cm<sup>2</sup> square was drawn on graph paper, and the mirror array was calibrated to display the square in the center of each hemisphere. Using ImageJ software, multiple variables were quantified, including distance of the inside edge of the square to the midline (X<sub>L</sub> or X<sub>R</sub>), height offset (Y), and the sizes of each square's length and width (X<sub>i</sub>, Y<sub>i</sub>). Bottom. These parameters were used to calculate angles of adjustment needed to be made to the mirrors such that alignment could occur. Additionally, a set of matrix transformations could be defined to normalize the areas of the squares. This transformation would functionally create two images of the same effective size from the two different perspectives.

**Conclusions/action items:**

This entry describes some of the math that can be used for both the testing of the device as well as calibration. Specifically, we defined the change in angle required to focus the mirrors in the horizontal as well as vertical directions. We also defined a matrix transformation that can be used to normalize the area projected from both hemispheres, regardless of whether a reflection occurs. This work will be useful when entering phase 2 of the project in which we apply transformation to the image to create a coherent projection for the user.





## 2022/10/10 Reserach on Anaglyph Filters

---

Sam Neuman - Oct 12, 2022, 12:07 AM CDT

**Title:** Reserach on Anaglyph Filters

**Date:** 10 October 2022

**Content by:** Sam

**Present:** Sam

**Goals:** Document my notes while reading a paper about creating and implementing anaglyph filters to 2D images

**Content:**

- Anaglyph filters are the red/blue filters used at movie theaters to create the perception of 3 dimensions. They require the user to wear red/blue glasses for it to work.

-This is based off of certain Gestalt principles (which are very abstract) and the idea that the brain takes the information perceived by the two eyes at the same time and tries to combine them into a meaningful image.

-The shadows produced by the filter have pixels whose RGB values are the sum of the original image's pixels RGB values and the RGB value of the desired color for the filter (i. e. blue).

-Thus, for us to use this, pixel/voxel math would be needed. We're not sure how difficult this would be for the computer to do in real-time.

-One way to do this, actually, is tensor algebra in which a tensor contains information about the pixel location, RGB values, and transformed shadow RGB values. This is a very bad way to do this but easy to code.

- Often 2D images don't contain any information about the 3D information. Actually this is entirely rare. Some AI can use computer vision principles to take a bad 'guess' as to the relative distances of objects in the image.

-Based on this lack of information alone, it is unlikely we would be able to use an anaglyph filter meaningfully.

-AI would be possible long-term, but not for our project. This paper gives a good way of how to do it. We also would need server-quality computer power for a few weeks and hours of surgery footage to make this work.

[1] E. Dubois, "A projection method to generate anaglyph stereo images," *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, 2001, pp. 1661-1664 vol.3, doi: 10.1109/ICASSP.2001.941256.

**Conclusions/action items:**

This paper gave a nice review of how anaglyph filters work and a mathematical background of how to make our own. The way to do this (most commonly used by professionals) is to create a tensor that contains all the information needed for the transformation, including the pixel/voxel original position, RGB values, and color change RGB values. What we didn't realize is that the anaglyph filter has variations in shadow size and strength based on the relative position of the image in the close/far direction (owing to the fixed distance of human pupils at around 63 mm). This could be discerned by some sort of AI framework, but it isn't feasible for our project. The anaglyph filter is really cool but simply not realistic for our project this semester.



## 2022/11/14 Research on VR Environment Coding

---

Sam Neuman - Dec 14, 2022, 10:58 PM CST

**Title:** Research on VR Environment Coding

**Date:** 14 November 2022

**Content by:** Sam

**Present:** Sam

**Goals:** Understand coding strategies to push livestreamed video to VR headset

**Content:**

There are two strategies to push videos from a smartphone camera to a VR headset. It is possible to directly bluetooth connect, but this requires the purchase of a compatible phone, and bluetooth connection is often less stable and reliable than wifi.

- Source code can be found that can be run on an Android phone to establish a secure server connection.
- Source code can also be developed via Unity software to recognize the IP address of the floating-point identity of the phone
- Within unity, a domain can be established corresponding to the IP of the smartphone such that the VR headset recognizes this IP during startup
- Videos from the internet showed this floating-point IP address to be maintained in the memory of the Oculus headset over multiple startup trials.

**Conclusions/action items:**

A major challenge of our design at its current stage is to establish a connection between the user's smartphone and Oculus VR headset. One solution can be used that involves source code for both the android smartphone and VR headset that allows them to recognize each other. When uploaded through Unity software, this IP identification is retained in the memory of the VR headset long-term over multiple startup trials. Our next steps, informed by this research, will be to write the scripts for both the phone and VR headset, incorporate them into unity software, and begin troubleshooting.



## 2022/09/30 - Design ideas to combine hardware and software design

HAOCHEN WANG - Oct 12, 2022, 10:31 AM CDT

**Title:** Design ideas to combine hardware design and software design

**Date:** 09/30/22

**Content by:** Haochen Wang

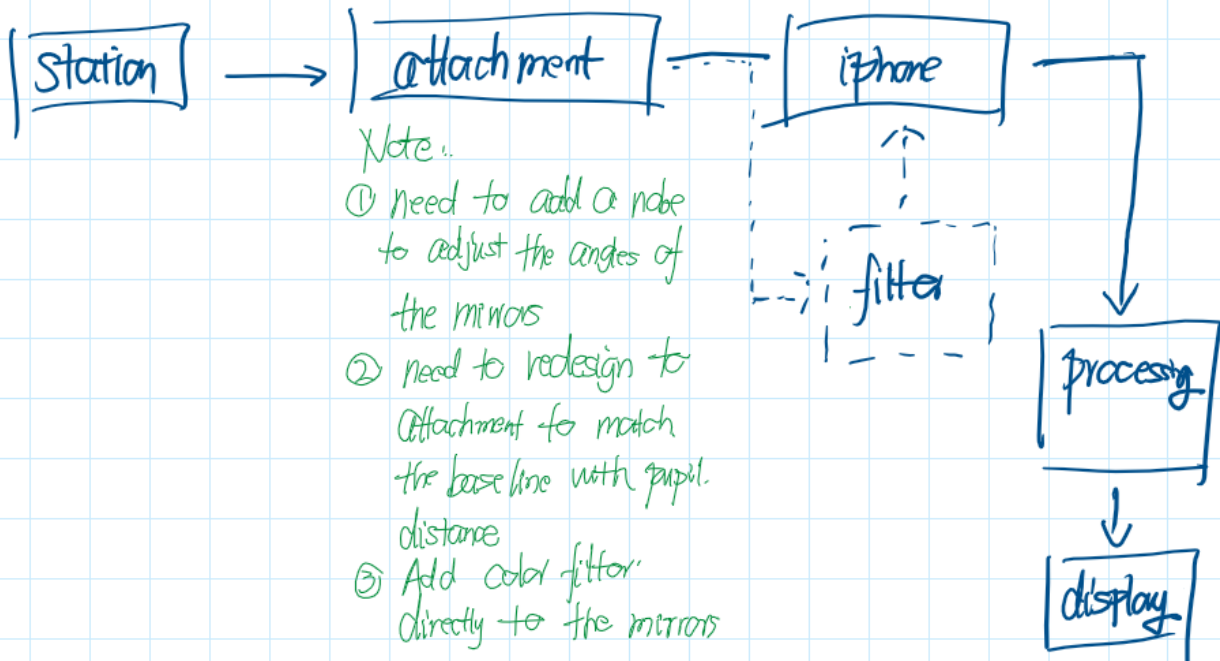
**Present:** N/A

**Goals:** To provide possible solutions for combining designs from previous years

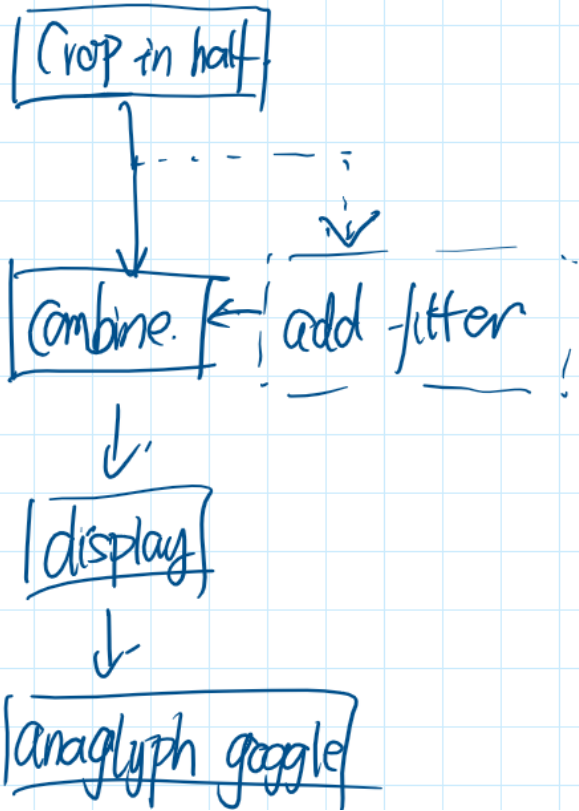
**Content:**

### Design matrix

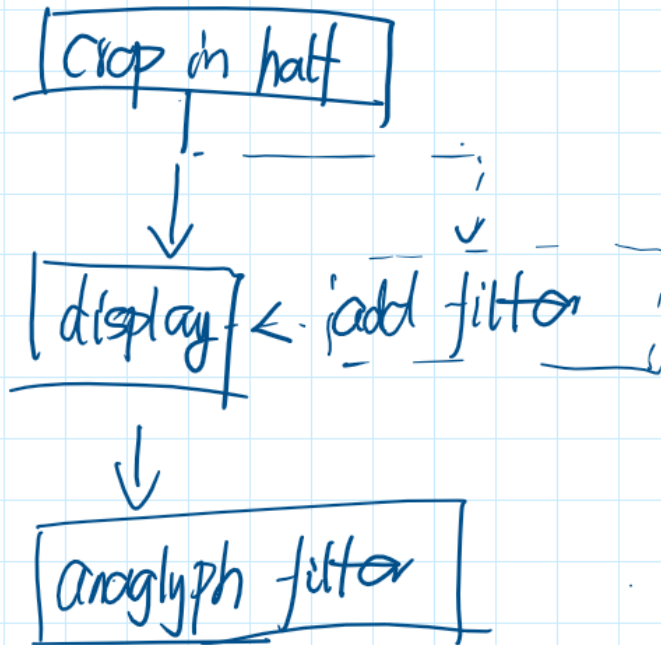
Friday, September 30, 2022 1:58 PM



① combined view



② shutter:



Note: the display here is an alternative between left and right eye view with  $\text{fps} \geq 30$

Note: dotted line for adding filter can be either implemented as a physical filter attached to mirrors in the smartphone attachment, or it can be set as one of the video processing speed in python script

Conclusions/action items:

Discuss these ideas during the advisor meeting and team mates



## 2022/10/05 - Python script

---

HAOCHEN WANG - Oct 12, 2022, 12:13 AM CDT

**Title:** Python Script 1.0

**Date:** 10/05/22

**Content by:** Haochen Wang

**Present:** N/A

**Goals:** To develop the first python script for video processing

**Content:**

```
import cv2
import sys
import numpy as np

cap = cv2.VideoCapture(r'C:\Users\Wang-\Desktop\BME400\test.mp4')

while(cap.isOpened()):
    ret, frame = cap.read()

    // add processing here

    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

**Conclusions/action items:**

Tried loading the microsurgery practice video from last semester. There seemed to be high consumption of cpu/graphic card power in my laptop (used for 3 years). Need to have a closer look into the method calls to reduce computation requirement, which is to leave space for actual video processing.



**# Version 4****# Note: Adopted from [Anaglyph 3D Stereo Imaging with PIL](#)****# Note: not working, requires further format conversion of each frame. Transitioned to Unity development**

```
import subprocess
import sys
import time
from UnityEngine import GameObject, PrimitiveType
subprocess.check_call([sys.executable, '-m', 'pip', 'install', 'numpy'])
subprocess.check_call([sys.executable, '-m', 'pip', 'install', 'opencv-python'])
```

```
import cv2
import numpy as np
```

```
# implement time tracking function
_start_time = time.time()
```

```
def tic():
    global _start_time
    _start_time = time.time()
```

```
def tac():
    t_sec = round(time.time() - _start_time)
    (t_min, t_sec) = divmod(t_sec,60)
    (t_hour,t_min) = divmod(t_min,60)
    print('Time passed: {}hour:{}min:{}sec'.format(t_hour,t_min,t_sec))
```

```
# read the video from specified path
cap = cv2.VideoCapture(1)
```

```
while(cap.isOpened()):
    # tic();
    # extract the frames
    ret, frame = cap.read()
```

```
if frame is not None:
    # read the size of the image, dimensions[2] is channel
    dimensions = frame.shape
    height = dimensions[0]
    width = dimensions[1]
    cut = int(width/2)
```

```
# crop the frame
leftView = frame[0:height, 0:cut]
```



```
rightView = frame[0:height, cut:width]

cv2.imshow('leftView',leftView)
cv2.imshow('rightView',rightView)

# tac();

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

cap.release()
cv2.destroyAllWindows()

# output to disk, default to no
dump_pipeline = "";

# raw video quality
# dump_pipeline = 'orig. ! queue ! filesink location=capture.raw'

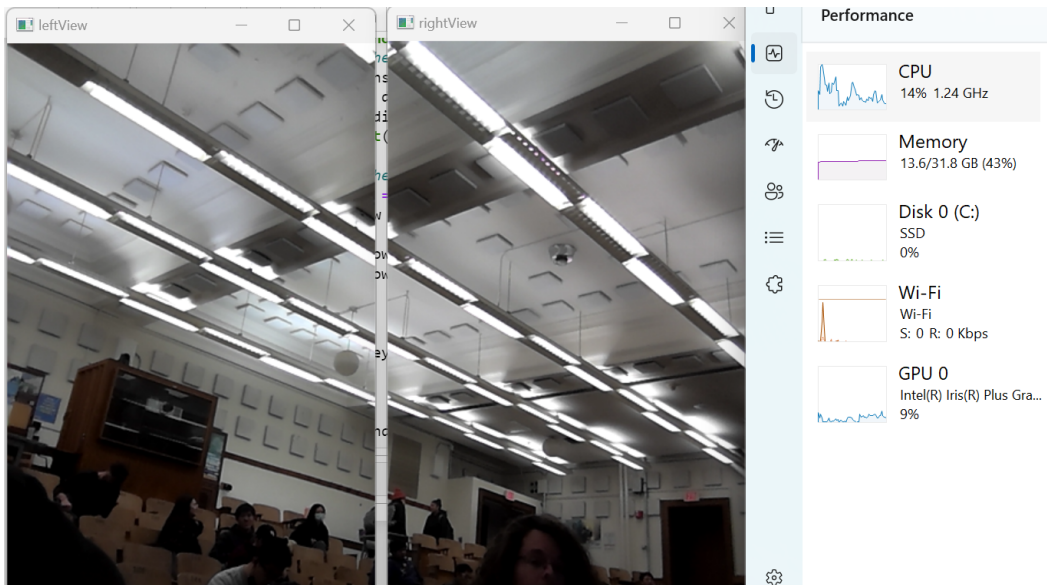
# intermediate video quality
# dump_pipeline = 'orig. ! jpegenc ! avimux ! queue ! filesink location=capture.mov'

# low video quality
# dump_pipeline = 'orig. ! jpegenc idct-method=2 quality=45 ! avimux ! queue ! filesink location=capture.mov'

default_scale = 100
viewports = 0
input_path = None

~~~~~

# Version 3
# Note: enabled default camera from the laptop
# Note2: enabled external camera from the smartphone
```




---

## # Version 2

# Note 1: open cv has bugs in displaying frames, so switched to a new method for image display

# Note 2: added cropping function and separated views for display

# Note 3: added processing time tracking function

```
import cv2
import sys
import numpy as np
import time
```

```
# implement time tracking function
```

```
_start_time = time.time()
```

```
def tic():
```

```
    global _start_time
```

```
    _start_time = time.time()
```

```
def tac():
```

```
    t_sec = round(time.time() - _start_time)
```

```
    (t_min, t_sec) = divmod(t_sec,60)
```

```
    (t_hour,t_min) = divmod(t_min,60)
```

```
    print('Time passed: {}hour:{}min:{}sec'.format(t_hour,t_min,t_sec))
```

```
# read the video from specified path
cap = cv2.VideoCapture(r'C:\Users\Wang-\Desktop\BME400\test.mp4')

while(cap.isOpened()):
    tic();
    # extract the frames
    ret, frame = cap.read()

    if frame is not None:
        # read the size of the image, dimensions[2] is channel
        dimensions = frame.shape
        height = dimensions[0]
        width = dimensions[1]
        cut = int(width/2)

        # crop the frame
        leftView = frame[0:height, 0:cut]
        rightView = frame[0:height, cut:width]

        cv2.imshow('leftView',leftView)
        cv2.imshow('rightView',rightView)

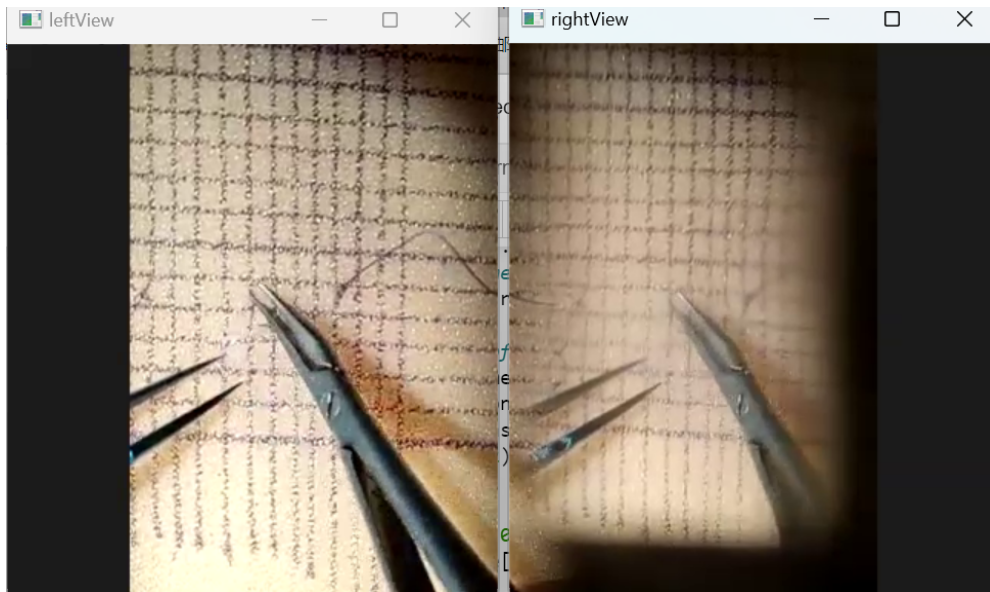
        tac();

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

---

Display results:



---

### #Version 1

```
import cv2
import sys
import numpy as np
```

```
cap = cv2.VideoCapture(r'C:\Users\Wang-\Desktop\BME400\test.mp4')
```

```
while(cap.isOpened()):
    ret, frame = cap.read()
```

```
    # add processing here
```

```
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

```
cap.release()
cv2.destroyAllWindows()
```

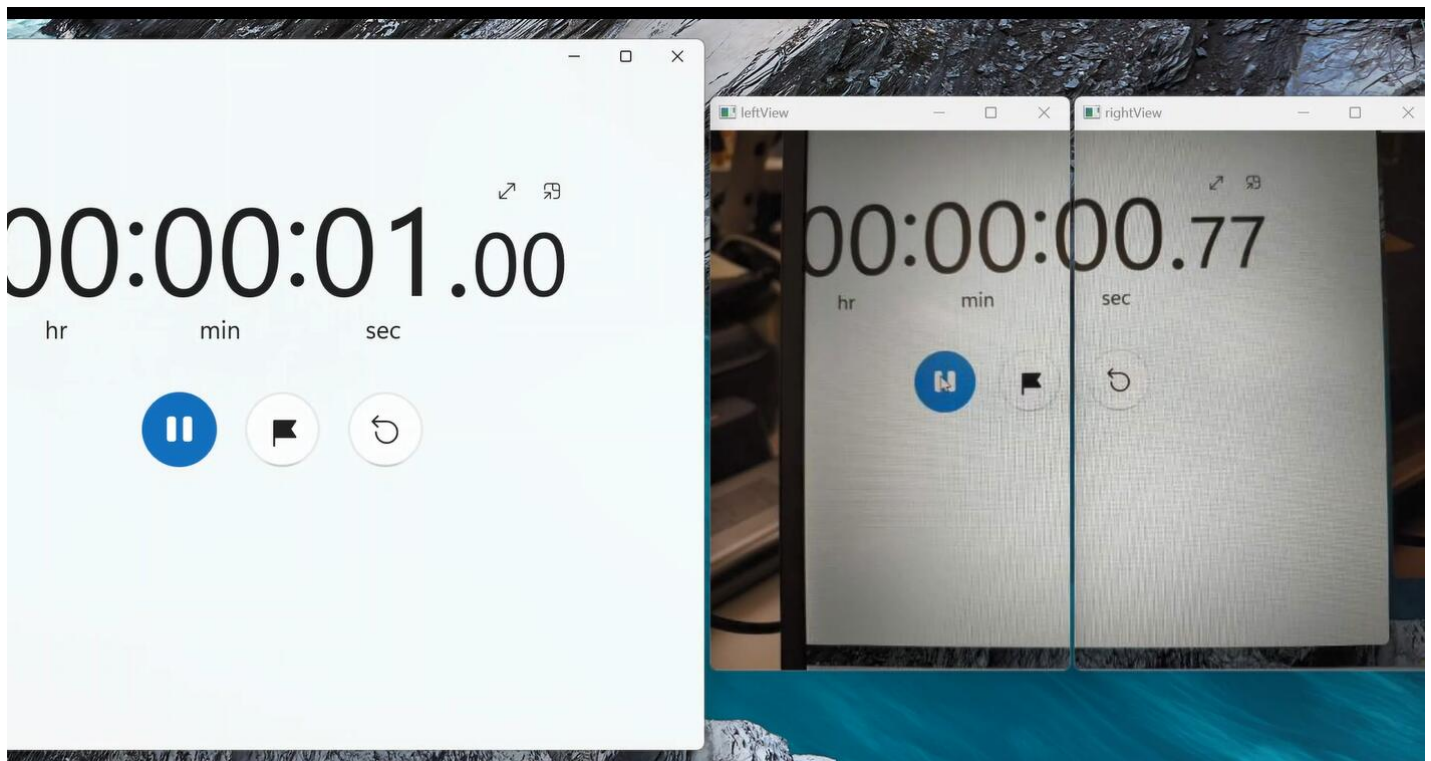


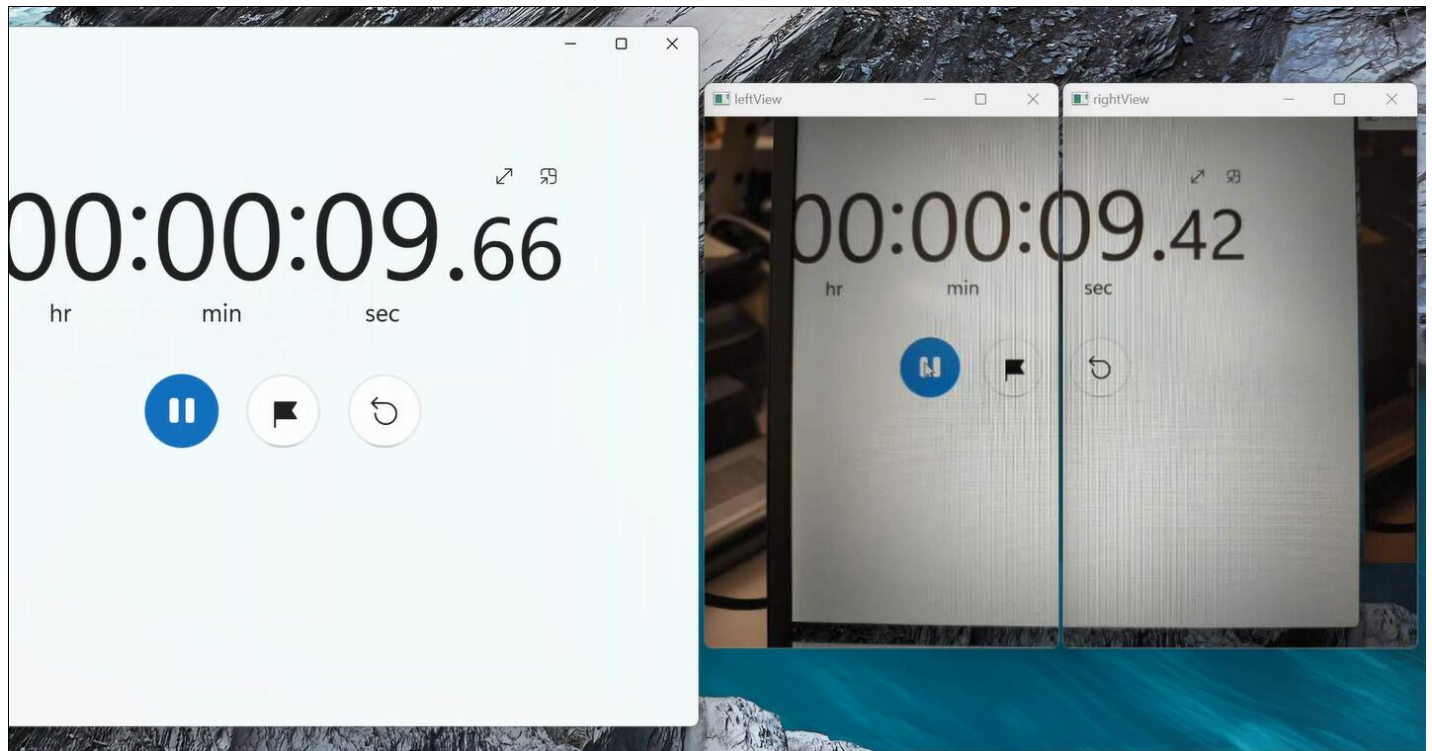
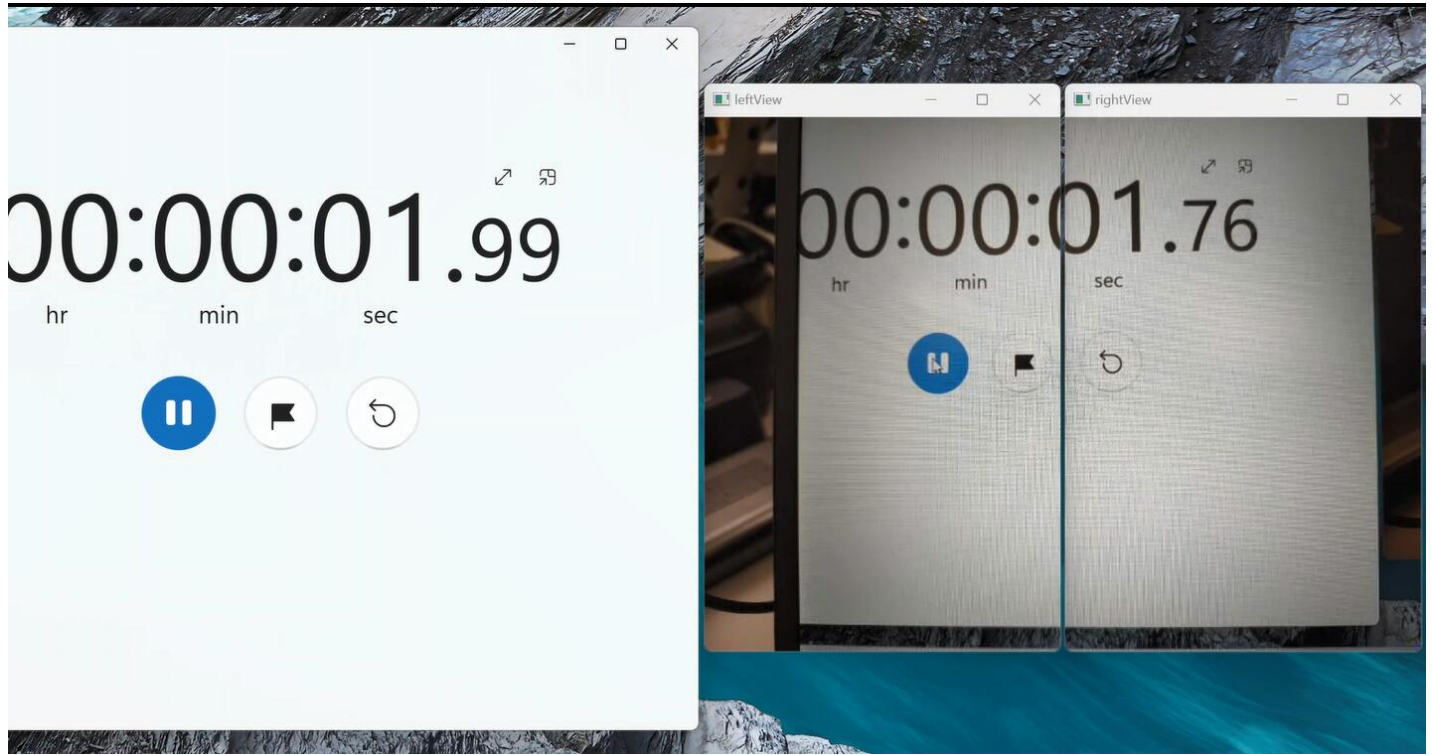
**Title: Python Script 1.0****Date:** 10/20/22**Content by:** Haochen Wang**Present:** N/A**Goals:** To test out time delay due to video processing**Content:**Procedure:

1. Have the smartphone connected to the laptop via USB connection, such that the laptop can recognize the smartphone as an external camera
2. Start the python script and use smartphone as the video capture device
3. Have the smartphone capture real time video for a stopwatch on the laptop, and record the split view along with the actual stopwatch via laptop screen capture
4. Then sample several frames from the screen capture, calculate the time delay between actual stop watch time and the transmitted real time video from the python code

Results

For all screenshots below, the left window is the actual stop watch, and the right one is the transmitted real time video processed by OpenCV





```

In [3]: import cv2
import sys
import numpy as np
import time

# implement time tracking function
_start_time = time.time()

def tic():
    global _start_time
    _start_time = time.time()

def tac():
    t_sec = round((time.time() - _start_time)
                 (t_hour, t_min) = divmod(t_sec, 60)
                 (t_hour, t_min) = divmod(t_min, 60)
                 print('Time passed: {}hour: {}min: {}sec'.format(t_hour, t_min, t_sec))

# read the video from specified path
cap = cv2.VideoCapture('C:\Users\Wang\Desktop\BME400\test.mp4')

while(cap.isOpened()):
    tic()
    # extract the frames
    ret, frame = cap.read()

    # read the size of the image, dimensions[2] is channel
    dimensions = frame.shape
    height = dimensions[0]
    width = dimensions[1]
    cut = int(width/2)

    # crop the frame
    leftView = frame[0:height, 0:cut]
    rightView = frame[0:height, cut:width]

    cv2.imshow('leftView', leftView)
    cv2.imshow('rightView', rightView)

    tac()

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec
Time passed: 0hour: 0min: 0sec: 00Sec

```

[Download](#)

**BME400\_ -\_ver.2.ipynb (229 kB)** 4 versions of python script

```

In [1]: import cv2
import sys
import numpy as np
import time

# implement time tracking function
_start_time = time.time()

def tic():
    global _start_time
    _start_time = time.time()

def tac():
    t_sec = round((time.time() - _start_time)
                 (t_hour, t_min) = divmod(t_sec, 60)
                 (t_hour, t_min) = divmod(t_min, 60)
                 print('Time passed: {}hour: {}min: {}sec'.format(t_hour, t_min, t_sec))

# read the video from specified path
cap = cv2.VideoCapture(3)

while(cap.isOpened()):
    tic()
    # extract the frames
    ret, frame = cap.read()

    # read the size of the image, dimensions[2] is channel
    dimensions = frame.shape
    height = dimensions[0]
    width = dimensions[1]
    cut = int(width/2)

    # crop the frame
    leftView = frame[0:height, 0:cut]
    rightView = frame[0:height, cut:width]

    cv2.imshow('leftView', leftView)
    cv2.imshow('rightView', rightView)

    # exit()

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

In [ ]:

```

[Download](#)

**BME400\_ -\_ver.3.ipynb (2.48 kB)** 4 versions of python script



```

In [8]: import cv2
import sys
import numpy as np
import time
import PCL, PCL.Image, PCL.ImageOps, PCL.ImageInstance

In [9]: # implement time tracking function
_start_time = time.time()

def tic():
    global _start_time
    _start_time = time.time()

def toc():
    t_sec = round(time.time) - _start_time
    (L_hrs, L_min) = divmod(t_sec, 60)
    (L_min, L_sec) = divmod(L_min, 60)
    print("Time passed: {}hour: {}min: {}sec".format(L_hrs, L_min, L_sec))

In [10]: # read the video from specified path
cap = cv2.VideoCapture(1)

while(cap.isOpened()):
    # tic()
    # extract the frame
    ret, frame = cap.read()

    # frame is not None
    # read the size of the image, dimensions[2] is channel
    dimensions = frame.shape
    height = dimensions[0]
    width = dimensions[1]
    cut = int(width/2)

    # crop the frame
    leftView = frame[0:height, 0:cut]
    rightView = frame[0:height, cut:width]
    leftConvert = cv2.cvtColor(leftView, cv2.COLOR_BGR2RGB)
    rightConvert = cv2.cvtColor(rightView, cv2.COLOR_BGR2RGB)
    leftLoad = PCL.Image.FromArray(leftConvert)
    rightLoad = PCL.Image.FromArray(rightConvert)
    left = PCL.ImageOps.colorize(leftLoad, (0, 0, 0), (0, 255, 255))
    right = PCL.ImageOps.colorize(rightLoad, (0, 0, 0), (255, 0, 0))

    cv2.imshow("leftView", left)
    cv2.imshow("rightView", right)
    blend = PCL.Image.blend(left, right, 0.5)
    cv2.imshow("Merge", blend)

    # toc()

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

```

[Download](#)

**BME400\_-\_ver.4\_discarded\_switched\_to\_VR\_set\_design\_.ipynb (6.47 kB)** 4 versions of python script

```

In [1]: import subprocess
import sys
import time
from UnityEngine import GameObject, PrimitiveType
subprocess.check_call([sys.executable, '-e', 'pip', 'install', 'numpy'])
subprocess.check_call([sys.executable, '-e', 'pip', 'install', 'opengl-py3win32'])

import cv2
import numpy as np

In [2]: # implement time tracking function
_start_time = time.time()

def tic():
    global _start_time
    _start_time = time.time()

def toc():
    t_sec = round(time.time) - _start_time
    (L_hrs, L_min) = divmod(t_sec, 60)
    (L_min, L_sec) = divmod(L_min, 60)
    print("Time passed: {}hour: {}min: {}sec".format(L_hrs, L_min, L_sec))

In [3]: # read the video from specified path
cap = cv2.VideoCapture(1)

while(cap.isOpened()):
    # tic()
    # extract the frame
    ret, frame = cap.read()

    # frame is not None
    # read the size of the image, dimensions[2] is channel
    dimensions = frame.shape
    height = dimensions[0]
    width = dimensions[1]
    cut = int(width/2)

    # crop the frame
    leftView = frame[0:height, 0:cut]
    rightView = frame[0:height, cut:width]

    cv2.imshow("leftView", leftView)
    cv2.imshow("rightView", rightView)

    # toc()

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

In [4]: # output to disk, default to no
img_pipeline = ''

# the video quality
# dump baseline = 'cmd: /q /w /f /b /l /c /d /e /f /g /h /i /j /k /l /m /n /o /p /q /r /s /t /u /v /w /x /y /z'

```

[Download](#)

**BME400\_-\_ver.4a.ipynb (3.34 kB)** 4 versions of python script

```
In [13]: import cv2
import sys
import numpy as np

In [14]: cap = cv2.VideoCapture('C:\Users\wang\Desktop\BME400\test.mp4')

while(cap.isOpened()):
    ret, frame = cap.read()

    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

-----
error:
Input in [14], in <cell line: 3>():
3 while(cap.isOpened()):
4     ret, frame = cap.read()
--> 11     cv2.imshow('frame',frame)
12     if cv2.waitKey(1) & 0xFF == ord('q'):
13         break

error: OpenCV(4.5.0) D:\opencv-python\opencv-python\opencv\modules\highgui\src\windows_w_capi.cpp: error: (-215:Assertion failed) size.width>0 && size.height>0 in function 'cv::imshow'
```

[Download](#)

**BME400\_-\_ver.1.ipynb (2.98 kB)** 4 versions of python script



## 2022/11/17 - Unity with OpenCV

---

HAOCHEN WANG - Nov 27, 2022, 8:36 PM CST

**Title:** Unity project with OpenCV

**Date:** 11/17/22

**Content by:** Haochen Wang

**Present:** N/A

**Goals:** To create a unity project using Open CV

**Preparation:**

Streaming platform: Unity, with "OpenCV in unity"

C coding: VS Studio

**Content:**

- After some initial research (sites need to be added), it would be easier to use C language version of Open CV and directly push the video to the VR set via Unity with C#.
- Tutorial for C# programming in unity
  - <https://unity.com/how-to/learning-c-sharp-unity-beginners>
- Tutorial for streaming with unity: <https://youtu.be/gB1F9G0JXOo>
  -
- Tutorial for streaming with OpenCv: <https://youtu.be/lXvt66A0i3Q>

**Conclusions/action items:**

**Date:** 11/25/22

**Preparation:**

Streaming platform: Unity, with "OpenCV in unity" and XR packages

C# coding: VS Studio

**Content:**

- Switch camera with button
  - <https://youtu.be/vtM-kqaTUSo>
- Create app for oculus
  - <https://headjack.io/knowledge-base/a-step-by-step-guide-for-oculus-app-lab-submissions/>
- Build project for oculus
  - <https://www.youtube.com/watch?v=pNYY1JsS7tY>
  - <https://jeffrafter.com/oculus-quest/>
  - <https://www.youtube.com/watch?v=gGYtahQjmWQ>
  - <https://www.youtube.com/watch?v=NWBc0y95obc&t=647s>
  - <https://circuitstream.com/blog/oculus-quest-2-setup/>
- AR for unity
  - (In Mandarin) <https://zhuanlan.zhihu.com/p/454706529>
- Separate view
  - [https://www.reddit.com/r/Unity3D/comments/trafak/how\\_to\\_render\\_different\\_content\\_for\\_each\\_eye/](https://www.reddit.com/r/Unity3D/comments/trafak/how_to_render_different_content_for_each_eye/)
- 

**Date:** 11/27/22

**Preparation:**

Streaming platform: Unity, with "OpenCV in unity" and XR packages

C# coding: VS Studio

**Content:**

- Rendering camera to Unity
  - <https://www.youtube.com/watch?v=ZV5eejYG6NI>

**Date:** 12/02/22

**Preparation:**

Streaming platform: Unity, with "OpenCV in unity" and XR packages

C# coding: VS Studio

**Content:**

***CameraScript.cs used in Image Acquisition Phase***

```
using System.Collections;
using System.Collections.Generic;
using TMLPro;
using UnityEngine;
using UnityEngine.UI;
```

```
public class CameraScript : MonoBehaviour
{
```

```
    int currentCamIndex = 1;
    //int frameWidth;
```

```
    private static WebCamTexture tex;
```

```
    public RawImage display;
```

```
    public TextMeshProUGUI startStopText;
```

```
    public void SwapCam_Clicked()
```

```
    {
        if (WebCamTexture.devices.Length > 0)
        {
            currentCamIndex += 1;
            currentCamIndex %= WebCamTexture.devices.Length;
```

```
            // If tex is not null
            // Stop the webcam and start new cam
            if (tex != null)
            {
                StopWebcam();
```

```

        StartStopCam_Clicked();
    }
}

```

```

public void StartStopCam_Clicked()
{
    if (tex != null) // Stop the camera
    {
        StopWebcam();
        startStopText.text = "Start Camera";
    }
    else // Start the camera
    {
        WebCamDevice device = WebCamTexture.devices[currentCamIndex];
        tex = new WebCamTexture(device.name);
        // frameWidth = tex.requestedWidth / 2;
        display.texture = tex;

        tex.Play();
        startStopText.text = "Stop Camera";
    }
}

```

```

public void StopWebcam()
{
    display.texture = null;
    tex.Stop();
    tex = null;
}

```

```

public static WebCamTexture Tex
{
    get { return tex; }
    set { tex = value; }
}

```

### **CameraControl.cs to be used in Video Processing**

```

using System.Collections;
using System.Collections.Generic;
using TMLPro;
using UnityEngine;
using UnityEngine.UI;

```

```
using OpenCvSharp;
using System;
```

```
public class CameraControl : MonoBehaviour
{
    public static int frameHeight;
    public static int frameWidth;
    int currentCamIndex = 1;
    public static WebCamTexture tex;
    public RawImage displayL;
    //public RawImage displayR;
    public TextMeshProUGUI startStopText;
    public Mat imageL;
    public Mat imageR;
    public Mat image;
    // private CameraControlRight controlRight = new CameraControlRight();
```

```
public void SwapCam_Clicked()
{
    if (WebCamTexture.devices.Length > 0)
    {
        currentCamIndex += 1;
        currentCamIndex %= WebCamTexture.devices.Length;
```

```

        // If tex is not null
        // Stop the webcam and start new cam
        if (tex != null)
        {
            StopWebcam();
            StartStopCam_Clicked();
        }
    }
}
```

```
public void StartStopCam_Clicked()
{
    if (tex != null) // Stop the camera
    {
        StopWebcam();
        startStopText.text = "Start Camera";
    }
    else // Start the camera
    {
        WebCamDevice device = WebCamTexture.devices[currentCamIndex];
        tex = new WebCamTexture(device.name);
        image = OpenCvSharp.Unity.TextureToMat(tex);
```

```

    if (frameWidth == 0 || frameHeight == 0)
    {
        frameWidth = tex.requestedWidth;
        frameHeight = tex.requestedHeight;
    }
    imageL = image.SubMat(0, frameHeight, 0, frameWidth / 2);
    imageR = image.SubMat(0, frameHeight, frameWidth / 2, frameWidth);

    displayL.texture = OpenCvSharp.Unity.MatToTexture(imageL);
    // displayR.texture = OpenCvSharp.Unity.MatToTexture(imageR);

    //CameraControlRight.CameraShowRight(this.imageR);
    startStopText.text = "Stop Camera";
}
}

```

```

public void StopWebcam()
{
    displayL.texture = null;
    //CameraControlRight.displayR.texture = null;
    // displayR.texture = null;
    tex.Stop();
    tex = null;
}
}

```

#### **D. CameraControlRight.cs to be used in Video Processing**

```

using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.UI;
using OpenCvSharp;

```

```

public class CameraControlRight : CameraControl
{
    public RawImage displayR;

    public void CameraShowRight()
    {
        displayR.texture = OpenCvSharp.Unity.MatToTexture(imageR);
    }
}

```



}

HAOCHEN WANG - Dec 14, 2022, 4:32 PM CST



[Download](#)

**WorkStation-Ver2.zip (1.56 GB LA S3)** Six versions of Unity project, each with different setting on the scene or the code. The currently adopted one is Version 1, using Oculus VR setting for camera plus simple camera control without OpenCV processing

HAOCHEN WANG - Dec 14, 2022, 4:33 PM CST



[Download](#)

**WorkStation-Ver3.zip (832 MB LA S3)** Six versions of Unity project, each with different setting on the scene or the code. The currently adopted one is Version 1, using Oculus VR setting for camera plus simple camera control without OpenCV processing

HAOCHEN WANG - Dec 14, 2022, 4:35 PM CST



[Download](#)

**WorkStation-Ver4.zip (714 MB LA S3)** Six versions of Unity project, each with different setting on the scene or the code. The currently adopted one is Version 1, using Oculus VR setting for camera plus simple camera control without OpenCV processing

HAOCHEN WANG - Dec 14, 2022, 4:37 PM CST



[Download](#)

**WorkStation-Ver5.zip (1.2 GB LA S3)** Six versions of Unity project, each with different setting on the scene or the code. The currently adopted one is Version 1, using Oculus VR setting for camera plus simple camera control without OpenCV processing

HAOCHEN WANG - Dec 14, 2022, 4:37 PM CST



[Download](#)

**WorkStation-Ver6.zip (113 MB)** Six versions of Unity project, each with different setting on the scene or the code. The currently adopted one is Version 1, using Oculus VR setting for camera plus simple camera control without OpenCV processing



[Download](#)

**WorkStation-Ver1.zip (3.41 GB LA S3)** Six versions of Unity project, each with different setting on the scene or the code. The currently adopted one is Version 1, using Oculus VR setting for camera plus simple camera control without OpenCV processing

**Title: Image and Video Processing in Python****Date:** 09/21/22**Content by:** Haochen Wang**Present:** N/A**Goals:** To learn how to do image processing with python**Content:**Website: <https://new.pythonforengineers.com/blog/python-for-scientists-and-engineers/>

1. Display an image
2. Blur and grayscale
  1. used gaussian blur
3. Edge detection
4. Count objects
  1. This seems to be the algorithm used by ImageJ
  2. Blue -> edge detection -> count
5. Face detection
  1. with cascades, from rough sections of blocks to more detailed check
  2. involves machine learning
6. \*\*\* use webcam
  1. Code:

```
1. import cv2
import sys

cascPath = "haarcascade_frontalface_default.xml"
face_cascade = cv2.CascadeClassifier(cascPath)
```

```
2. if len(sys.argv) < 2:
    video_capture = cv2.VideoCapture(0)
else:
    video_capture = cv2.VideoCapture(sys.argv[1])
```

```
3. while True:
    # Capture frame-by-frame
    ret, image = video_capture.read()
```

```
4. ret, image = video_capture.read()

    if not ret:
        break

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (30,30)

    )

    #print("The number of faces found = ", len(faces))

    for (x,y,w,h) in faces:
        cv2.rectangle(image, (x,y), (x+h, y+h), (0, 255, 0), 2)

    cv2.imshow("Faces found", image)
```

```
5. if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

```
6. # When everything is done, release the capture  
video_capture.release()  
cv2.destroyAllWindows()
```

```
7. python webcam_face_detect.py
```

```
8. python webcam_face_detect.py webcam.mp4
```

2. 3 => read returns the actual video frame read from each loop and a return code (which tells if we are running out of frames)

3. 5=> press key q to quit

4. 6=>> clean up

5. 7=> use webcam

6. 8=> view the video

## 7. Motion detection

1. Code

```

1.  #!/usr/bin/python

import cv2
import sys
import numpy as np

if len(sys.argv) < 2:
    video_capture = cv2.VideoCapture(0)
else:
    video_capture = cv2.VideoCapture(sys.argv[1])

2. # Read two frames, last and current, and convert current to gray.
ret, last_frame = video_capture.read()
ret, current_frame = video_capture.read()
gray = cv2.cvtColor(current_frame, cv2.COLOR_BGR2GRAY)

3. i = 0
while(True):
    # We want two frames- last and current, so that we can calculate the
    different between them.
    # Store the current frame as last_frame, and then read a new one
    last_frame = current_frame

4. ret, current_frame = video_capture.read()
gray = cv2.cvtColor(current_frame, cv2.COLOR_BGR2GRAY)

5. # Find the absolute difference between frames
diff = cv2.absdiff(last_frame, current_frame)

6. # If difference is greater than a threshold, that means motion
detected.
if np.mean(diff) > 10:
    print("Achtung! Motion detected.")

7. # Display the resulting frame
cv2.imshow('Video',diff)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
# When everything done, release the capture
video_capture.release()
cv2.destroyAllWindows()

```

2. 5=> note the use of cv2.absdiff to compare differences

### Conclusions/action items:

1. check import numpy and a good IDE (like ancorda)
- 2.



**Title:** Photo inpainting python tutorial

**Date:** 09/23/22

**Content by:** Haochen Wang

**Present:** N/A

**Goals:** To understand the python algorithm for converting photos into 3D-like photos

**Content:**

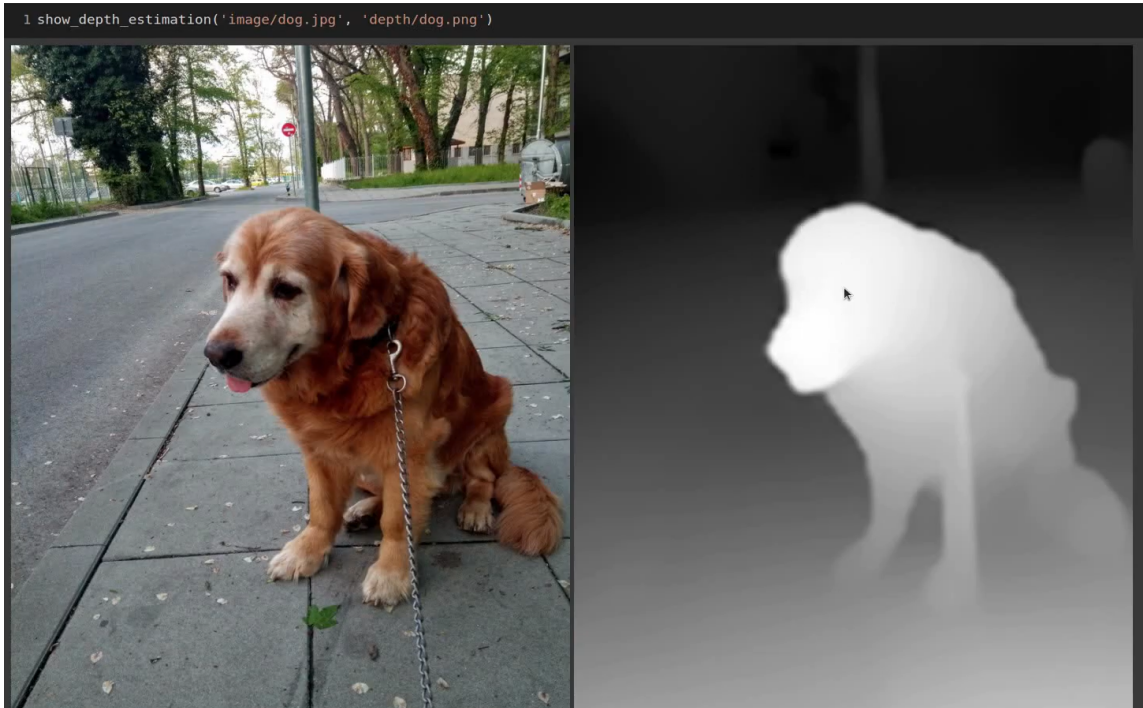
Video link: <https://youtu.be/x54GdkpW9JY>

Related article: 3D photography using context-aware layered depth inpainting, by Meng-Li Shih et al.

1. Install dependencies

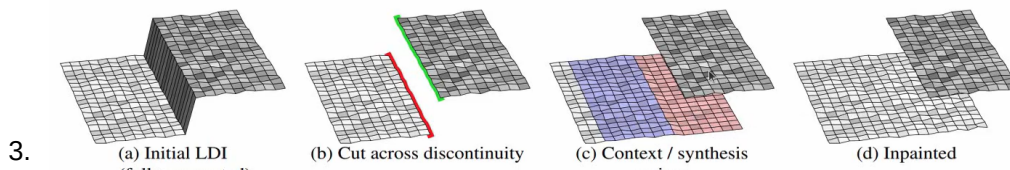
1. !pip install -q visp==0.6.4
2. !pip install -q moviepy==1.0.2
3. !pip install -q transforms3d==0.3.1
4. !pip install -q network==2.3
5. !pip install -q -U watermark
- 6.
7. !sudo apt install sed

2. the Midas package on gitHub includes functions for estimating depths (from the Intel-isl group)





- 2.
3. whiter pixel in the depth detection means that the depth is lowered (closer to the camera)
4. Can use MiDaS to obtained RGB-D images
  1. Could the mirrored view be used as the D-detection?
  2. The image quality from the mirror is really not as good, so probably can just serve as D-factor adding onto the direct view, which has better focusing and resolution



### Conclusions/action items:

Consider how D-factor should be obtained and incorporated to generate the 3D effect

The processing time from the video demo, for one frame only, takes about 10 seconds. So a possible solution is to generate the depth map at the beginning of the microsurgery practice, with background, station and the skin/blood vessel only. Then keep using this depth map as the depth for all following frames.

\*\*\*Check out the original paper from the authors of the code

# 2022/09/29-3D photography using context-aware layered depth inpainting

HAOCHEN WANG - Sep 30, 2022, 10:25 AM CDT

**Title:** 3D photography using context-aware layered depth inpainting

**Date:** 09/29/22

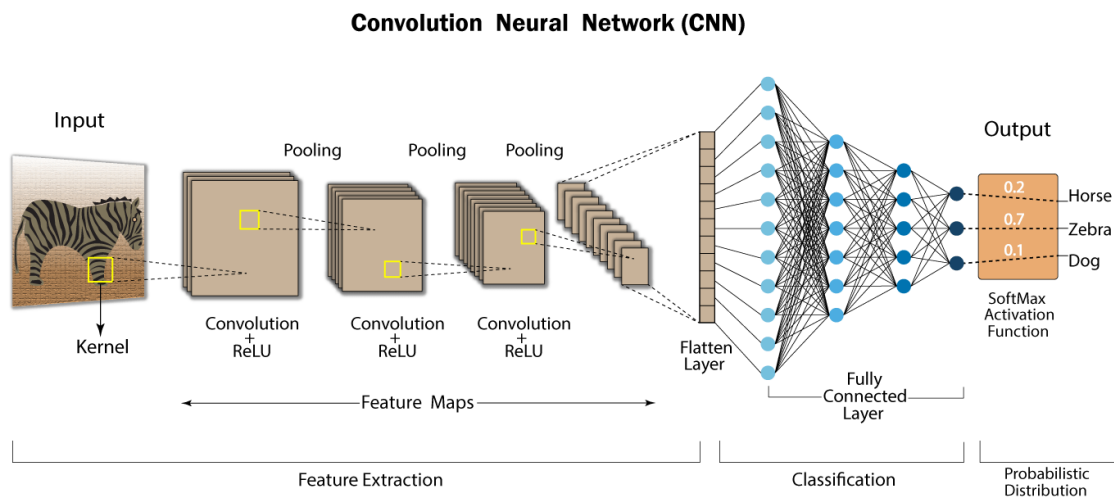
**Content by:** Haochen Wang

**Present:** N/A

**Goals:** To understand the mechanism of inpainting

**Content:**

1. Facebook 3D photos require capturing a single snapshot with a dual lens camera phone, which essentially provides an RGB-D input image
  1. uses layered depth image (LDI), converted into a light-weight mesh representation for rendering
  2. isotropic diffusion algorithm for inpainting colors
  3. but result in overly smooth results
2. Multi-plane representation (MPI), synthesized from the small-baseline dual camera stereo input, but produces artifacts on sloped surfaces
  1. also memory and storage inefficient
  2. costly
3. The paper uses algorithm based on spatially adaptive context regions, designing to the local depth-complexity of input and generates a varying number of layers across the image
4. View synthesis: light field, multi-plane image representation (fixed depth discretization, so not working for sloped surfaces), layered depth images [[Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 231–242. ACM, 1998.](#)] can locally adapt to any depth complexity
5. Image based rendering: need sufficiently large baselines --> learning-based blending, soft 3D reconstruction, handling reflection, relighting, and reconstructing mirror and glass surfaces
6. Learning based view synthesis: most current work require pre-trained network
7. Image inpainting: CNN-based architectures (convolutional neural network)
  1. Found this image on google to explain that it is



2.

8.



**Conclusions/action items:**

check these two articles (lowkey 3D capture)

<https://dl.acm.org/doi/10.1145/3197517.3201384>

<http://visual.cs.ucl.ac.uk/pubs/casual3d/>



## 2022/10/17 - Application of Deliverable in Ophthalmology

---

HAOCHEN WANG - Oct 17, 2022, 10:29 PM CDT

**Title:** Application of Deliverable in Ophthalmology

**Date:** 10/17/22

**Content by:** Haochen Wang

**Present:** N/A

**Goals:** To provide an example of how our deliverable can be utilized in other fields

**Content:**

[P. Singh, S. K. Bergaal, P. Sharma, T. Agarwal, R. Saxena, and S. Phuljhele, "Effect of induced anisometropia on stereopsis and surgical tasks in a simulated environment," vol. 69, no. 3, 2021, \[Online\]. Available:](https://journals.lww.com/ijo/Fulltext/2021/03000/Effect_of_induced_anisometropia_on_stereopsis_and_surgical_tasks_in_a_simulated_environment)

[https://journals.lww.com/ijo/Fulltext/2021/03000/Effect\\_of\\_induced\\_anisometropia\\_on\\_stereopsis\\_and.19.aspx](https://journals.lww.com/ijo/Fulltext/2021/03000/Effect_of_induced_anisometropia_on_stereopsis_and.19.aspx)

["Eyesi Surgical," Haag-Streit Simulation. \[Online\]. Available: https://www.vrmagic.com/medical-simulators/eyesi-surgical. \[Accessed: 17-Oct-2022\].](https://www.vrmagic.com/medical-simulators/eyesi-surgical)

- The first paper uses Eyesi evaluation system to evaluate the performance of trainees. The authors uses a simulating platform that allows trainees to practice ophthalmology surgery with an interactive model through microscopes.
- With our deliverable, there is no need for waiting the availability of microscopes, and the evaluation can be performed promptly

**Conclusions/action items:**

Share this finding with other team members



## 2022/10/02- Potential Ideas

---

ALEXANDER VAZQUEZ - Oct 12, 2022, 2:13 PM CDT

**Title:** Flow of Data

**Date:** 10/2/22

**Content by:** Alex

**Present:** Alex

**Goals:** Come up with some potential flow ideas

**Content:**

**Previous Team Idea:**

- Image split by single pair of mirrors
- Image captured by iPhone camera
- Video feed sent over zoom call to user

**Alternative Design 1:**

- Image split by single pair of mirrors spaced rough human pupillary distance
- Image captured by phone camera
- Feed sent wirelessly to compute device where post processing occurs
- Device directly connected to display, method of switching between monitor display or google cardboard display

**Alternative Design 2:**

- Image split by single pair of mirrors spaced rough human pupillary distance
- Image captured by iPhone
- Code executed to add anaglyph filter over captured image
- Send to a display via a cord to reduce latency
- User wears anaglyph filter glasses to perceive depth

**Alternative Design 3:**

- Video capture done with iPhone Lidar sensor
- Use raw 3D image and stream/hardware to display directly
  - Alternatively we could use the result to create a depth mask give depth to the video feed
- Add any additional computer or other step to facilitate quick image processing

**Conclusions/action items:**

There are various ways that the clients conditions can be met. More research into the topics can give us insight into which is the most effective/realistic solution.



## 2022/09/12- Background Project Research

---

ALEXANDER VAZQUEZ - Oct 12, 2022, 8:59 AM CDT

**Title:** Project Catch Up

**Date:** 9/12/22

**Content by:** Alex

**Present:** Alex

**Goals:** Go through past material to get a better understanding of the project scope and what needs to be done.

**Content:**

**Document 1: Final Report**

**Abstract/Introduction**

- **Problem:** Medical students travel to larger hospitals to train with micro surgical equipment. This standard for training medical students is inconvenient and has become more difficult due to travel restrictions.

**Background**

- Surgical Microscopes:
  - Offer high resolution and depth perception
  - High magnification
  - Expensive, \$200k-\$1m
  - Sources Viewed: [1][2][3]

**Preliminary Designs**

- Grey pro resin: High accuracy, rigid material, inexpensive (Winner)
- Tough 1500 Resin: High accuracy, weaker material, not ideal for holding mirrors.
- Acrylic (Laser Cut): Highest accuracy, strongest material, expensive, difficult to assemble

**Final Design**

- Grey Pro Resin
- 1 mirror pair for image capture
- Image capture projected a outside display, or second smartphone in google cardboard

**Results**

- The final prototype completed several goals of the project; depth perception, high resolution/quality image, visible sutures
- Image not seamless, high delay in relaying picture
- Inaccurate pupil distance, adjustments could improve functionality of stereoscopic view

**Thoughts:**

- Main improvements that we should make are to the design and orientation of the mirrors/ housing
  - Simulate human pupillary distance
  - Ensure design is sturdy to prevent shifting
- Improve system of merging images to create a more seamless picture

- Find a streaming method that offers lower latency
- **New ideas:**
  - Different Methods of displaying picture (Multiple output options for flexibility)
  -

**Document 2: PDS**

- **Budget:** 500\$ not including; iPhone, stand, and attachment for camera
- Other criteria similar to what was in the report

**Conclusions/action items:**

**I think I have a better idea of the goals of this project and where the team got last year. I think we can make meaningful improvements to the design this semester and provide a viable design for a training setting. Next, we should meet as a group to discuss how w want to proceed this semester and what design ideas we have.**



## 2022/09/24-iPhone Design Research

---

ALEXANDER VAZQUEZ - Oct 12, 2022, 10:06 AM CDT

**Title:** iPhone Design Understanding

**Date:** 9/25/22

**Content by:** Alex

**Present:** Alex

**Goals:** Gain a better understanding of the design of the iPhone camera system as well as how these features can be leveraged for our design needs.

**Content:**

**iPhone 13 Pro Camera:**

- Offers 12MP camera with three different camera types. Telephoto, Wide and Ultra Wide
- Portrait mode that offers Depth Control
- Can record up to 4k 60 fps
- Offers a very powerful photography device in a small package

Biggest benefit from the iPhone is that it is a small but very powerful package. Its connectability makes it easier to move the video feed to different sources. The powerful camera and the Lidar sensor could prove to be useful.

Source:

“iPhone 13 Pro - Technical Specifications.” [https://support.apple.com/kb/SP852?locale=en\\_US](https://support.apple.com/kb/SP852?locale=en_US) (accessed Oct. 24, 2022).

- iPhone Lidar detectors can be used to develop a 3D maps of objects or rooms:
  - Could be used to create a depth mask over the image.
- Laser scans millions of points and return the information to the sensor
- \*Look into Lidar sensor applications: “Photogrammetric” and “Point Cloud”
- May be limited to pictures depending on process time

Source:

I. Aslan and N. Polat, “Availability of Iphone 13 Pro Laser Data in 3D Modeling,” *Advanced LiDAR*, vol. 2, no. 1, Art. no. 1, Mar. 2022.

**Conclusions/action items:**

**The iPhone camera is indeed powerful enough to capture the quality image that our client needs. The main concern is the sense of depth. A split view can do this but the Lidar sensor on the iPhone could also prove to be a beneficial tool.**



**Title:**

**Date:**

**Content by:**

**Present:**

**Goals:**

**Content:**

**Conclusions/action items:**