BME Design-Fall 2020 - Jason Wang
Complete Notebook

PDF Version generated by

Xiaoxuan Ren

on

Dec 09, 2020 @02:06 PM CST

Table of Contents

Project Information 2
Team contact Information 2
Project description 3
Team activities 4
Client Meetings 4
09/11/2020 - Client Meeting 4
09/18/2020 - Client Meeting 6
09/29/2020 - Client Meeting 7
Advisor Meetings 8
09/11/2020 - Advisor Meeting 8
09/18/2020 - Advisor Meeting 9
09/25/2020 - Advisor Meeting 10
Jiong Chen 11
Research Notes 11
Biology and Physiology 11
2020/09/09 - Global Health Microsurgery Training With Cell Phones 11
2020/09/10 - Can Smartphones Be Used to Perform Video-Assisted Microanastomosis? An Experimental Study 13
Competing Designs 15
2020/09/14 - RICOH THETA VR Camera 15
Design Ideas 17
2020/09/14 Reality A 17
2020/09/20 - 3D image conversion 18
2020/09/24 3D Edison sonic wonder 21
2020/10/13 retriving data from camera 22
Swift Coding and Debugging 24
2020/12/05 - AVFoundation 24
2020/12/07 - Type Conversion Error 25
2020/12/08 - Installing Error 26
2020/12/08 - Testing Of The Output 27
2020/12/08 - Final Version of Swift Code 29
Jason Wang 32
Research Notes 32
Biology and Physiology 32
Global Health Microsurgery Training With Cell Phones 32
Can Smartphones Be Used to Perform Video-Assisted Microanastomosis? An Experimental Study 33
Protocol 35
MM51 Brochure 36
Orbeye 38
3D Glasses 40
Design Ideas 41
Google Cardboard 41
Training Documentation 42
Green Shop Pass 42
Meetings 43
Advisor Meeting 43
09/11/2020 - Advisor Meeting 43

09/18/2020 - Advisor Meeting 45
09/25/2020 - Advisor Meeting 46
10/09/2020 - Advisor Meeting 47
10/16/2020 - Advisor Meeting 48
10/23/2020 - Advisor Meeting 49
10/30/2020 - Advisor Meeting 50
11/06/2020 - Advisor Meeting 51
11/13/2020 - Advisor Meeting 52
11/20/2020 - Advisor Meeting 53
Client Meeting 54
09/11/2020 - Client Meeting 54
09/18/2020 - Client Meeting 56
09/29/2020 - Client Meeting 57
10/09/2020 - Client Meeting 58
10/23/2020 - Client Meeting 59
11/06/2020 - Client Meeting 60
11/20/2020 - Client Meeting 61
12/04/2020 - Client Meeting 62
Team Meeting 63
09/11/2020 - Team Meeting 63
Martin Janiszewski 64
Research Notes 64
Biology and Physiology 64
2020/9/17 GoPro Cameras 64
2020/9/17 Lightning Cable Transfer Speeds 65
2020/9/25 IEEE Automatic Real-time 2D-to-3D Conversion for scenic views 66
2020/10/1 Heads-up 3D Microscopy: An Ergonomic and Educational Approach to Microsurgery 68
Code Implementation Research 70
2020/12/6 - anaglyph3d script 70
2020/12/6 - Anaglyph from One Image 73
2020/12/7 - Converting Video into Frames for Analysis 76
Design Ideas 79
Design Work: Digital Anaglyph 79
2020/12/6 - Basic Anaglyph Photo Conversion 79
2020/12/6 - one_image_anaglyph script 81
2020/12/7 - ExtractMovieFrames with Anaglyph Conversion Full Script 84
Xiaoxuan Ren 88
Research Notes 88
08/29/2020 background research 88
09/11/2020 real-time 3D rendering 90
09/14/2020 machine learning based Depth Image-Based Rendering (DIBR) algorithm 92
Design Ideas 94
09/27/2020 3D glass protocol 94
Xcode --display video 95
Xcode -- depth extraction 96
12/9/2020 Read Depth Data 98
2014/11/03-Entry guidelines 100
2014/11/03-Template 101

Team contact Information

Jason Wang - Sep 10, 2020, 10:08 PM CDT

Last Name	First Name	Role	E-mail	Phone	Office Room/Building
Thompkins	Willis	Advisor	tompkins@engr.wisc.edu	608-263-1581	
Shaffrey	Ellen	Client	ellencshaffrey@gmail.com		
Wang	Jason	Leader	jwang889@wisc.edu	920-313-0320	
Ren	Xiaoxuan	Communicator	xren37@wisc.edu		
Janiszewski	Martin	BSAC + BPAG	mjaniszewski@wisc.edu		
Chen	Jiong	BWIG	jchen672@wisc.edu		

Project description

Jason Wang - Oct 07, 2020, 3:50 PM CDT

Course Number: BME 400

Project Name: iPhone Virtual Reality Training Model for Microsurgical Practice

Short Name: iPhone VR

Project description/problem statement:

iPhones are used ubiquitously as training devices for emerging physicians. The iPhone has the camera capabilities that can provide magnification comparable to surgical microscopes used for vessel anastomoses. A previous attempt has been made to produce virtual reality glasses using a computer, two cell phones, and lightning cable connection. However, there is too much delay from this current model and we have been unable to design a simple streamlined model. The goal for this project is to design a simple iPhone-VR system to create a home microsurgery simulation tool that could be used as a resource for resident surgeons to practice microsurgery.

About the client: Group of microsurgical residents and doctors

09/11/2020 - Client Meeting

Jason Wang - Oct 07, 2020, 3:47 PM CDT

Title: Client Meeting

Date: 09/11/2020

Content by: Jason

Present: All

Goals:

Content:

- Client
    - Ellen - 2nd year resident
    - Dr. Poore
    - + a couple other not present
- iPhone
    - Has good resolution and magnification
    - Limitation through lack of depth perception
        - Possible Solution: VR Goggles
- Microsurgical microscopes are usually used, but expensive
    - Alternatives:
        - Exoscope
            - High resolution camera that projects to a monitor
        - Possibly iPhone/smartphone setup
- Model 1
    - iPhone + iPhone holder/stand + the basic camera app
        - ~$20-30 not including iPhone
    - iPhone connected to Macbook Pro
        - Using a lightning cable attached to both
    - Use quicktime player to do a live screenshare of the iPhone
    - Advantages
        - Magnification maintained
        - Easy to set up
        - Minimal delay between actual phone camera and the screenshare
        - Relatively inexpensive
    - Disadvantages
        - Unable to use the full screen of the macbook???
        - iPhone needs to be vertical for proper image orientation?
        - Need for an external light source
        - Lack of Depth Perception
    - Ideal Set up for end-product
- Model 2
    - iPhone + Reality A app?
    - Advantages
        - Allows you to change interpupillary distance
        - Provides a full screen image
        - Able to use the iPhone's built in light
        - Capable of recording
    - Disadvantages
        - No zoom/magnification capability
        - Apparently not user friendly
            - Not touch sensitive or Bad touch programming
- Model 3
    - iPhone + iPhone + Google Cardboard + Teamviewer/Duet
    - Advantages
        -
    - Disadvantages
        - Significant delay between actual camera and screenshare (.5-1s)
        - Decreased user friendliness

        - Multiple devices and connections
- Magnification of microscope anywhere from 2x to 25x
- Prefer more direct view, peripherals are not as important
- Prefer to try and use smartphone, but not necessary

Conclusions/action items:

09/18/2020 - Client Meeting

Jason Wang - Oct 07, 2020, 3:47 PM CDT

Title: Client Meeting

Date: 09/11/2020

Content by: Jason

Present: All

Goals:

Content:

- Discussed the design ideas that could be used for the project itself
- Asked clarifying question from the week prior
- Client
    - expressed interesting in purchasing 3d lens attachment for the iPhone
    - Acknowledged that the project doesn't need to follow the smartphone route, but would be prefered
    - Questioned if the google cardboard lenses could be modified for a longer focal length

Conclusions/action items:

09/29/2020 - Client Meeting

Jason Wang - Oct 07, 2020, 3:47 PM CDT

Title: Client Meeting

Date: 09/29/2020

Content by: Jason

Present: All

Goals:

Content:

- Followed up about the changing of the google cardboard lenses to increase focal length
    - Does not seem feasible as the size of the google cardboard would have to be increased
- Posture
    - Posture does not seem to be an issue as stated by the clients
    - Having a good quality image is more important
    - Prior to the surgery
        - Everything is prepped to the surgeons preferences
            - Camera zoom
            - hand position
            - resolution
            - camera position
- Main concern
    - Eliminate as many screens as possible
        - reduce the delay/latency between screens
-

Conclusions/action items:

09/11/2020 - Advisor Meeting

Jason Wang - Oct 07, 2020, 3:48 PM CDT

Title: Advisor Meeting

Date: 09/11/2020

Content by: Jason

Present: All

Goals:

Content:

- Look into
    - ipad pro LIDAR
    - App production
        - xcode and SwiftUI
    - Similar devices
- Make sure to get a lot of useful information from the client to know where to begin research and work on the project

Conclusions/action items:

09/18/2020 - Advisor Meeting

Jason Wang - Oct 07, 2020, 3:48 PM CDT

Title: Advisor Meeting

Date: 09/18/2020

Content by: Jason

Present: All

Goals:

Content:

- Talked about possible research pathways to follow
- Xiaoxuan mentioned neural networks
- Overall, team agreed to focus on researching VR glasses and 3D depth perception technologies

Conclusions/action items:

09/25/2020 - Advisor Meeting

Jason Wang - Oct 07, 2020, 3:48 PM CDT

Title: Advisor Meeting

Date: 09/25/2020

Content by: Jason

Present: All

Goals:

Content:

- Discussed the Design matrix ideas
    - Advisor concerned over posture of the microsurgeon while using the monitor from design 1
    - Discussed the use of 3d glasses and the different types
    - Discussed the presentation guidelines

Conclusions/action items:

2020/09/09 - Global Health Microsurgery Training With Cell Phones

Jiong Chen - Oct 06, 2020, 4:49 PM CDT

Title: Global Health Microsurgery Training With Cell Phones

Date: 2020/09/09

Content by: Jiong Chen

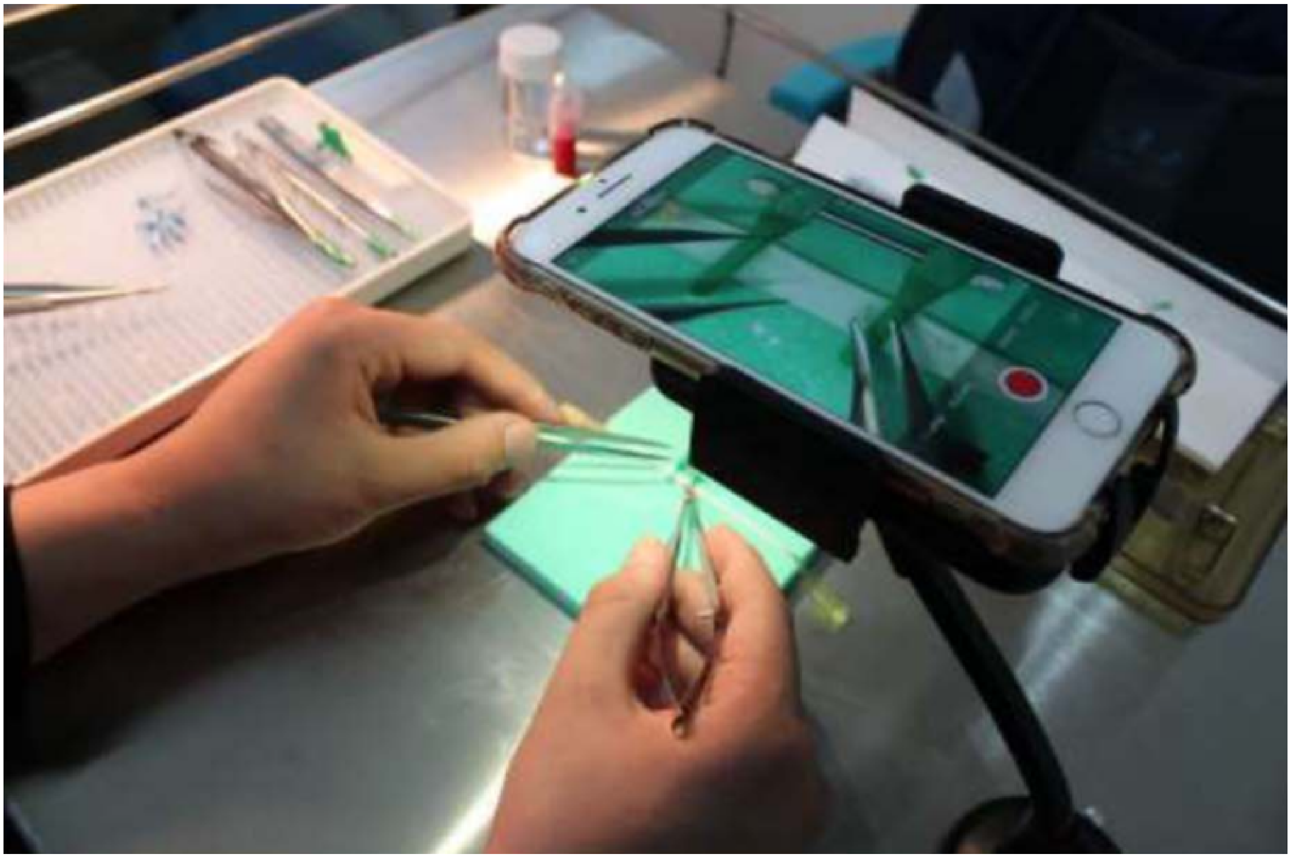
Present: NA

Goals: To read the paper provided by the client

Content:

This paper is written by Suzanne M. Inchauste from Stanford University Medical Center. The study used the camera function of the smart-phone to perform zoom capability as a mimic of loupe during surgical suturing.

Due to the lack of resources in some developing countries, it is hard for trainees receiving medical help training to get access to a microscope for practice purposes. In this case, the use of smartphone would hopefully accommodate this need as shown in the following picture from the article.



After training faculties with smartphones, a survey was conducted among the trainees:

TABLE 3. Results of Microscope Training Survey About Comparing Cell Phone and Loupes (n = 9)

How easy is it to set up the cell phone stand for microsurgical practice?				
Not easy	Slightly easy	Moderately easy	Easy	Very easy
1	1	2	5	
How satisfied are you using cell phone for microsurgical practice?				
Very dissatisfied	Dissatisfied	Neutral	Satisfied	Very satisfied
	1	1	7	
How likely are you to use the cell phone for microsurgical practice at home or work?				
Not likely	Slightly likely	Moderately likely	Likely	Very likely
5	1		3	
How would you rate the depth perception while using the cell phone for microsurgical practice?				
Not good	Slightly good	Moderately good	Good	Very good
2	2	3	1	1
How would you rate the clarity while using the cell phone for microsurgical practice?				
Not good	Slightly good	Moderately good	Good	Very good
2	1	2	3	1
How satisfied are you using loupes for microsurgical practice?				
Very dissatisfied	Dissatisfied	Neutral	Satisfied	Very satisfied
4			2	3
How likely are you to use loupes for microsurgical practice at home or work?				
Not likely	Slightly likely	Moderately likely	Likely	Very likely
1	1		3	4
How would you rate the depth perception while using loupes for microsurgical practice?				
Not good	Slightly good	Moderately good	Good	Very good
			5	4
How would you rate the clarity while using loupes for microsurgical practice?				
Not good	Slightly good	Moderately good	Good	Very good
			5	4
Do you prefer cell phone or loupes to practice?				
Cell phone			Loupes	
2			7	

Almost all of the trainees have improved their suturing skills with the smartphone-based practice. However, the user experience of the practice is not perfect. The biggest problem reported is the depth precision and the clarity of the vision. In addition, loupes are still more preferred than the smartphone.

Conclusions/action items:

The paper provided a good background on how important the problem is and how urgent this need is. Our project aims to develop 3-D VR-based training, so I think it will greatly improve the problem of precision and clarity than using a smartphone with flat screen.

2020/09/10 - Can Smartphones Be Used to Perform Video-Assisted Microanastomosis? An Experimental Study

Jiong Chen - Oct 06, 2020, 4:49 PM CDT

Title: Can Smartphones Be Used to Perform Video-Assisted Microanastomosis? An Experimental Study

Date: 2020/09/10

Content by: Jiong Chen

Present: NA

Goals: To read the information provided by client

Content:

The study in this article performed a similar idea from the Global Health Microsurgery Training With Cell Phones. Differences between those two studies are that this study conducted actual animal experiment while another study used suturing practice. In addition, the set up in this study includes a TV connected to the smartphone (as shown in the picture below from the article)

Figure 2. Video-system composed of smartphone and full HD TV set connected by Wi-Fi.

However, the results of this experiment were disappointing. When conducting a real animal experiment, due to the lack of stereoscopic view and insufficient resolution, both of the groups using the iPhone and Samsung failed to finish the surgery even in 3 hours, and the surgery is usually finished within half an hour.

Therefore, even though the smartphone provides a better image and focus than microscopes, it still has some fetal disadvantages that make the use of smartphones not feasible. Here are some advantages and disadvantages of smartphone-based assisting systems:

pros: better ergonomy, autofocus system, deployment of operating room personnel

cons: lack of stereoscope view, presence of cable near the setup, limitation of the surgical field by required support, and the need to acquire specific cameras.

Conclusions/action items:

The experiment is very similar to the previous one, and the problem is also very close - the resolution (clarity) and stereoscopic view. But I think with the VR technology used, the view from the assisting system will be much more close to real microscope.

2020/09/14 - RICOH THETA VR Camera

Jiong Chen - Oct 06, 2020, 4:49 PM CDT

Title: RICOH THERA VR Camera

Date: 2020/09/14

Content by: Jiong Chen

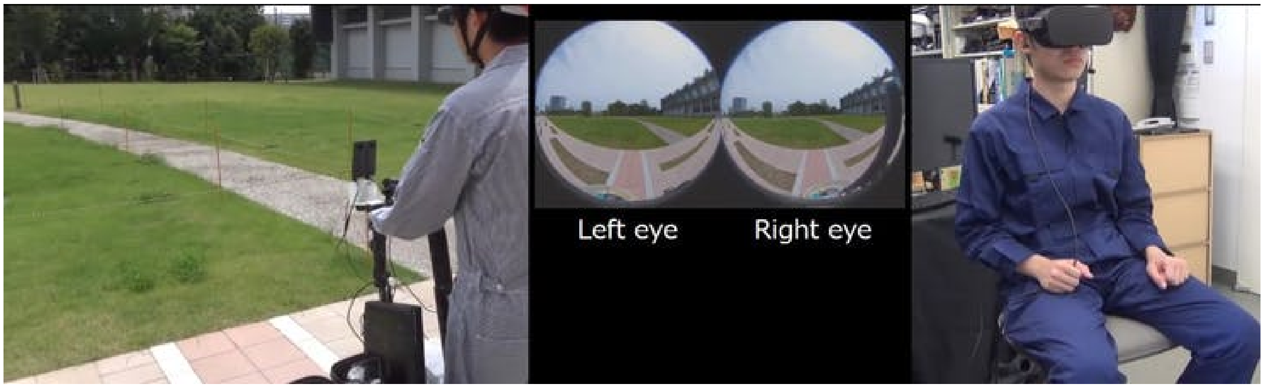
Present: NA

Goals: to find a similar product in the current market.

Content:

The RIICOH THERA VR Camera is a 360-degree camera that can live stream 4K video to a variety of VR headsets. The price for the current model is 379.99$ available online.

A research has conducted using the camera to control a Segway remotely, Two VR camera is attached on the Segway so that the environment around the Segway could be seen via a VR set. This study provided a successful result, which means that the delay should be small enough to allow the person to turn for different directions on the road. Additionally, the product can proffer up to 4K streaming with 30 Hz of freshing rate.



Conclusions/action items:

With such a camera, the delay could be almost neglected and as well as the depth perception. However, it is also relatively expensive and requires an additional VR set.

Article available online: https://www.hackster.io/news/stream-real-world-vr-scenes-to-your-headset-8a78dfedbb3e

2020/09/14 Reality A

Jiong Chen - Oct 06, 2020, 4:49 PM CDT

Title: Reality A

Date: 2020/09/14

Content by: Jiong Chen

Present: NA

Goals: record potential applications that could be used in the project

Content:

Reality Augmented (VR) Is The World's First "Stereoscopic" Virtual Augmented Reality Experience For iPhone. The app can directly forming VR compatible image with iPhone camera.

Still needs actual VR box to test resolution and depth perception

Conclusions/action items:

Available online: https://apps.apple.com/us/app/reality-augmented-vr/id954501500

2020/09/20 - 3D image conversion

Jiong Chen - Oct 06, 2020, 4:50 PM CDT

Title: 3D image conversion - anaglyph

Date: 2020/09/20

Content by: Jiong Chen

Present: NA

Goals: research anaglyph

Content:

Anaglyph Methods Comparison

True Anaglyphs

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0{,}299 & 0{,}587 & 0{,}114 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

- Dark image
- No color reproduction
- Little ghosting

Gray Anaglyphs

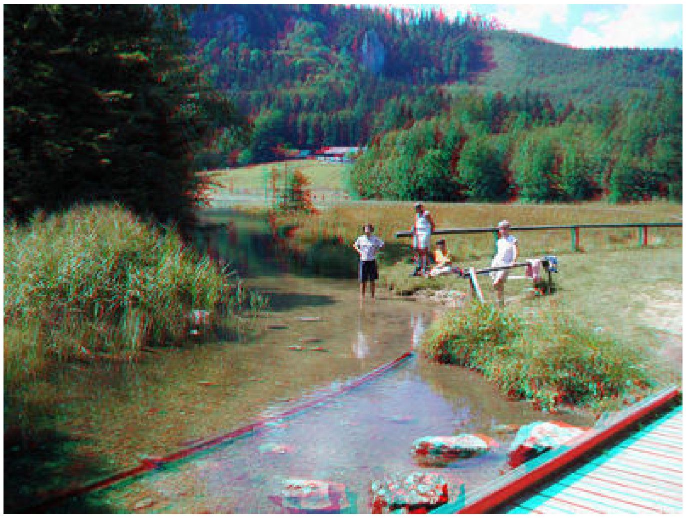
$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0{,}299 & 0{,}587 & 0{,}114 \\ 0{,}299 & 0{,}587 & 0{,}114 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

- No color reproduction
- More ghosting than true anaglyphs

Color Anaglyphs

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

- Partial color reproduction
- Retinal rivalry



Half Color Anaglyphs

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

- Partial color reproduction (but not as good as color anaglyphs)
- Less retinal rivalry than color anaglyphs

Optimized Anaglyphs

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0 & 0{,}7 & 0{,}3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

In addition, we applied a gamma correction (gamma value 1.5) to brighten up final red channel r_a. Stereoscopic Player implements a further improved algorithm which partially maps the red channels to green and blue before applying the above formula.

- Partial color reproduction (but not of red shades)
- Almost no retinal rivalry

When we talk about retinal rivalry, we just mean the retinal rivalry caused by brightness differences of colored objects. Of course there is an additional form of retinal rivalry, independently of the anaglyph method used: retinal rivalry caused by the different color channels perceived by left and right eye.

Conclusions/action items:

Potentially usable if we are going to coding for the device to achieve the real time 2D-3D conversion.

available at http://www.3dtv.at/knowhow/AnaglyphComparison_en.aspx

2020/09/24 3D Edison sonic wonder

Jiong Chen - Oct 06, 2020, 4:50 PM CDT

Title: 3D edison sonic wonder

Date: 2020/09/24 3D edison sonic wonder

Content by: Jiong Chen

Present: NA

Goals: research real time 3D conversion console

Content:

The proposed project package (product) is simple – it turns your ordinary TV that you watch on a daily basis, let it be an LCD display or projectors to produce 3D video. Yes, it's that easy. Simply plug-in an HDMI cable to your existing TV and connect the other end of HDMI to the palm-size console (3D Edison). 3D Edison System comprises of a 3D Console and a pair of 3D Active Shutter Glasses. All you need is your existing TV Set with an HDMI Input and a Media Source such as Standard TV Program me, DVD / Blu-ray Player and/or Gaming Consoles including your PC/MAC resolution@60Hz with an HDMI Output. This includes all Gaming Consoles + PC / Mac that are equipped with HDMI output. The 3D Edison Consoles accept BOTH 2D and 3D signals. This means, you can watch contents in 3D whether the playback content is encoded in 2D or 3D.

3D Edison - Connection

1. Connect the an HDMI Cable to the INPUT of 3D Edison Console and the OUTPUT of any media

source that is equipped with an HDMI Output.

2. Connect a 2nd HDMI Cable to the OUTPUT of 3D Edison Console and the INPUT of any 2D TV / Monitor / Project / Display.

3. Connect the supplied Power Cord to the 3D Edison Console and the AC Output of the Wall AC Outlet.

*** The 3D Edison Console accepts AC Voltage between 100 – 240VAC 50/60Hz***

Conclusions/action items:

The 3D Edison Consoles accept BOTH 2D and 3D signals. This means that we can watch contents in 3D whether the playback content is encoded in 2D or 3D. Additionally, the console allows changing the depth of the image, which exactly fulfilled our requirement.

available at https://xu-yun.com/

2020/10/13 retriving data from camera

Jiong Chen - Oct 13, 2020, 11:41 PM CDT

Title: retrieving data from the camera

Date: 2020/10/13

Content by: Jiong Chen

Present: NA

Goals: find ways to input video data from the camera to the laptop

Content:

```
% Create video input object.
vid=videoinput('winvideo',1,'YUY2_640x480');

% Set video input object properties for this application.
% Note that example uses both SET method and dot notation method.
set(vid,'TriggerRepeat',100);
vid.FrameGrabInterval = 1;

% Set value of a video source object property.
vid_src = getselectedsource(vid);
set(vid_src,'Tag','motion detection setup');
set(vid,'ReturnedColorSpace','rgb');

%to convert yuy2 image to rgb

% Create a figure window.
figure;

% Start acquiring frames.

start(vid)

while(vid.FramesAvailable <100)
data = getdata(vid);

imshow(data(:,:,:,2));

drawnow
% update figure window
%
end
stop(vid)
```

Conclusions/action items:

Sample code from online. This sample code uses a separate cam, but it has huge delays.

https://www.mathworks.com/matlabcentral/answers/124314-how-to-reduce-delay-in-video-processing

2020/12/05 - AVFoundation

Jiong Chen - Dec 08, 2020, 11:25 PM CST

Title: AVFoundation

Date: 2020/12/05

Content by: Jiong Chen

Present: NA

Goals: Research AVFoundation Framework

Content:

This page is from the apple official website (available online at https://developer.apple.com/documentation/avfoundation)**:**

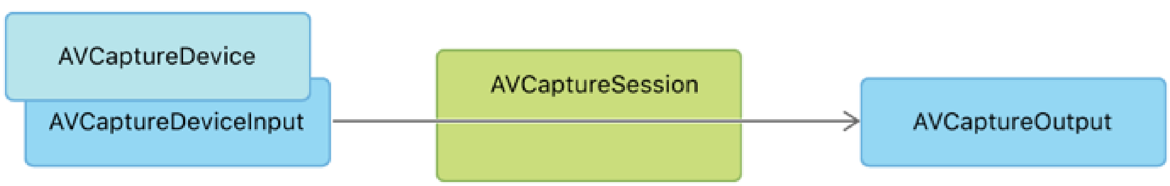
The AVFoundation Capture subsystem provides a common high-level architecture for video, photo, and audio capture services in iOS and macOS. Use this system if you want to:

- Build a custom camera UI to integrate shooting photos or videos into your app's user experience.
- Give users more direct control over photo and video capture, such as focus, exposure, and stabilization options.
- Produce different results than the system camera UI, such as RAW format photos, depth maps, or videos with custom timed metadata.
- Get live access to pixel or audio data streaming directly from a capture device.

Note

To instead let the user capture media with the system camera UI within your app, see `UIImagePickerController`.

The main parts of the capture architecture are sessions, inputs, and outputs: Capture sessions connect one or more inputs to one or more outputs. Inputs are sources of media, including capture devices like the cameras and microphones built into an iOS device or Mac. Outputs acquire media from inputs to produce useful data, such as movie files written to disk or raw pixel buffers available for live processing.



Important

Mac apps built with Mac Catalyst can't use the AVFoundation Capture classes. These apps should use `UIImagePickerController` for photo and video capture, instead.

Conclusion and thoughts:

The AVFoundation has built-in functions that establish the basic function of our application. It is the most important framework that we will have to learn. After we apply the AVFoundation, we will then add the zoom function into the input of the AVCaptureDevice to realize the magnification capability.

2020/12/07 - Type Conversion Error

Jiong Chen - Dec 08, 2020, 11:35 PM CST

Title: Type Conversion Error

Date: 2020/12/07

Content by: Jiong Chen

Present: NA

Goals: To record the most confusing error in this project so far

Content:

When I try to add the zooming function, I used the UIGesturePinchRecognizer. The error I got is "Cannot convert value of type 'UIGesturePinchRecognizer.Type' to expected argument type 'UIGesturePinchRecognizer'"

The reason why this happens is that the UIGesturePinchRecognizer is an object. So what passed into the method should also be the UIGesturePinchRecognizer object. However, with no object created, I was passing the type UIGesturePinchRecognizer into the method, which is abstract. In this case, the following code is necessary to solve this error:

let guesture = UIPinchGestureRecognizer(target:**self**, action:**#selector**(pinchAction))

self.view.addGestureRecognizer(guesture)

With the UIGesturePinchRecognizer object created, then it is okay to write

@IBAction func pinchAction(_ sender: UIPinchGestureRecognizer) {

}

and in this way the method will work correctly

2020/12/08 - Installing Error

Jiong Chen - Dec 08, 2020, 11:29 PM CST

Title: Installing Error

Date: 2020/12/08

Content by: Jiong Chen

Present: NA

Goals: Solving error message

'invalid code signature, inadequate entitlements or its profile has not been explicitly trusted by the user.'

Content:

When I try to test the application and install it on my iPhone, there is one error message says 'invalid code signature, inadequate entitlements or its profile has not been explicitly trusted by the user.'

This error is due to the trust permission on the iPhone for me, as the application developer. In this case, the solution is to go the Settings -> General -> Device Management, and then click trust the developer.

2020/12/08 - Testing Of The Output

Jiong Chen - Dec 08, 2020, 11:41 PM CST

Title: Testing Of The Output

Date: 2020/12/08

Content by: Jiong Chen

Present: NA

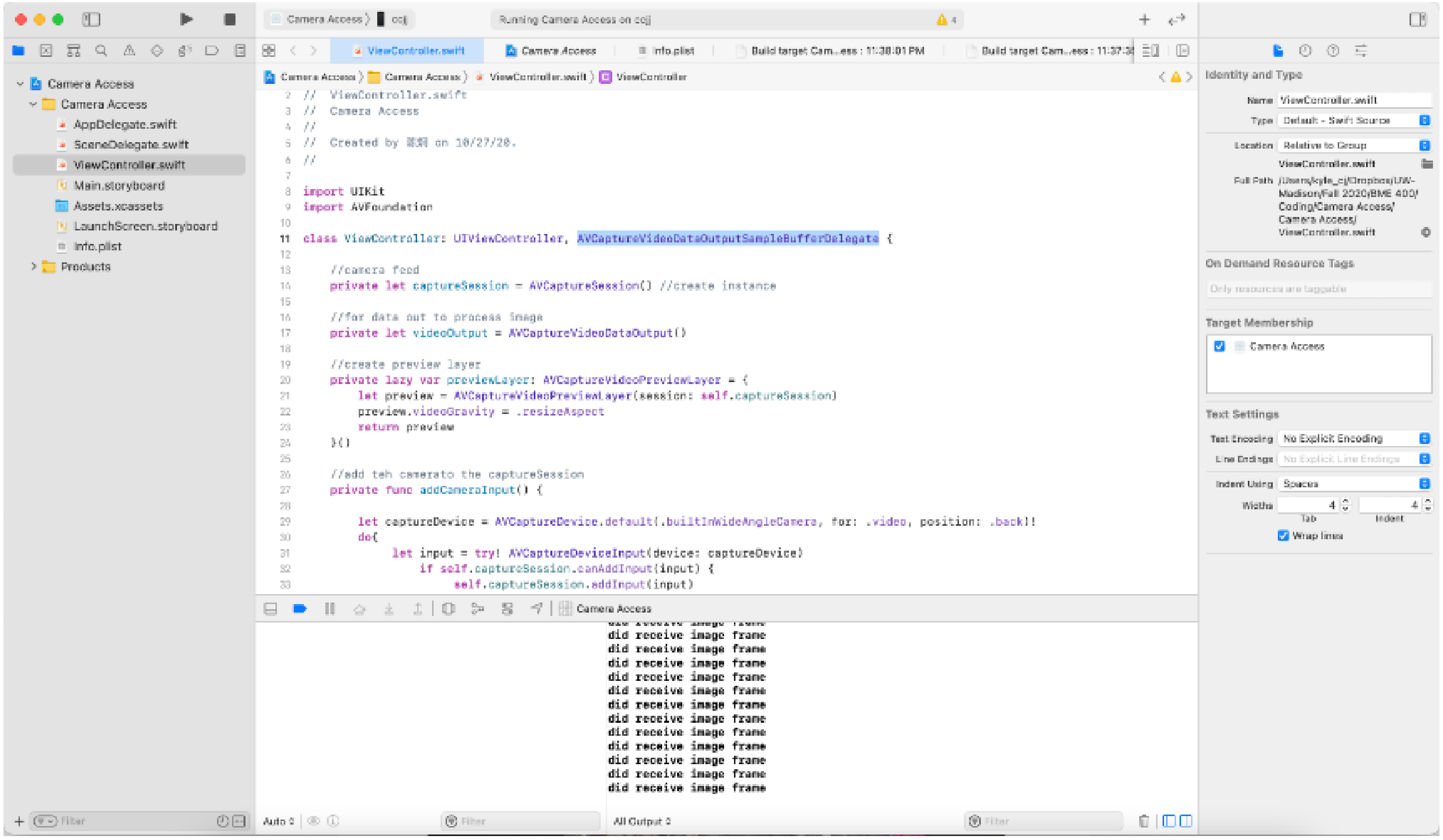
Goals: To create testing to check if the application is outputting the frame

Content:

In the method

```
func captureOutput(_ output: AVCaptureOutput,
                   didOutput sampleBuffer: CMSampleBuffer,
                   from connection: AVCaptureConnection) {
    guard let frame = CMSampleBufferGetImageBuffer(sampleBuffer) else {
        debugPrint("unable to get image from sample buffer")
        return
    }
    print("did receive image frame")
    // process image here
}
```

we are capturing the frames from the camera feed. If the application did receive the frame, the app will output the message "did receive image frame" from the XCode console. The testing results shown below



Running Camera Access on ccj
ViewController.swift
Camera Access
Info.plist
AppDelegate.swift
SceneDelegate.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Products
// ViewController.swift
// Camera Access
import UIKit
import AVFoundation
class ViewController: UIViewController, AVCaptureVideoDataOutputSampleBufferDelegate {
//camera feed
private let captureSession = AVCaptureSession() //create instance
//for data out to process image
private let videoOutput = AVCaptureVideoDataOutput()
//create preview layer
private lazy var previewLayer: AVCaptureVideoPreviewLayer = {
let preview = AVCaptureVideoPreviewLayer(session: self.captureSession)
preview.videoGravity = .resizeAspect
return preview
}()
//add teh camerato the captureSession
private func addCameraInput() {
let captureDevice = AVCaptureDevice.default(.builtInWideAngleCamera, for: .video, position: .back)!
do{
let input = try! AVCaptureDeviceInput(device: captureDevice)
if self.captureSession.canAddInput(input) {
self.captureSession.addInput(input)
did receive image frame
Identity and Type
Name ViewController.swift
Type Default - Swift Source
Location Relative to Group
Target Membership
Camera Access
Text Settings
Text Encoding No Explicit Encoding
Indent Using Spaces
Wrap lines
All Output

2020/12/08 - Final Version of Swift Code

Jiong Chen - Dec 08, 2020, 11:19 PM CST

Title: Final Version of Swift Code

Date: 2020/12/08

Content by: Jiong Chen

Present: NA

Goals: Record final version of the code

Content:

The application is a user interaction app, and all the code is written in the file ViewController.swift

```
//
//  ViewController.swift
//  Camera Access
//
//  Created by Jiong Chen on 10/27/20.
//

import UIKit
import AVFoundation

class ViewController: UIViewController, AVCaptureVideoDataOutputSampleBufferDelegate {

    //camera feed
    private let captureSession = AVCaptureSession() //create instance

    //for data out to process image
    private let videoOutput = AVCaptureVideoDataOutput()

    //create preview layer
    private lazy var previewLayer: AVCaptureVideoPreviewLayer = {
        let preview = AVCaptureVideoPreviewLayer(session: self.captureSession)
        preview.videoGravity = .resizeAspect
        return preview
    }()

    //add teh camerato the captureSession
    private func addCameraInput() {
```

```
        let captureDevice = AVCaptureDevice.default(.builtInWideAngleCamera, for: .video, position: .back)!
        do{
            let input = try! AVCaptureDeviceInput(device: captureDevice)
                if self.captureSession.canAddInput(input) {
                    self.captureSession.addInput(input)
                }
        }
        catch{
                print("Error")
        }
    }

    //add the preview layer to the app
    private func addPreviewLayer() {
        self.view.layer.addSublayer(self.previewLayer)
    }

    override func viewDidLayoutSubviews() {
        super.viewDidLayoutSubviews()
        self.previewLayer.frame = self.view.bounds
    }

    //set the image in format of 32BGRA
    private func addVideoOutput() {
        self.videoOutput.videoSettings = [(kCVPixelBufferPixelFormatTypeKey as NSString) : NSNumber(value: kCVPixelFormatType_32BGRA)] as
[String : Any]
        self.videoOutput.setSampleBufferDelegate(self, queue: DispatchQueue(label: "my.image.handling.queue"))
        self.captureSession.addOutput(self.videoOutput)
    }

    //receive camera frame
    func captureOutput(_ output: AVCaptureOutput,
                    didOutput sampleBuffer: CMSampleBuffer,
                    from connection: AVCaptureConnection) {
        guard let frame = CMSampleBufferGetImageBuffer(sampleBuffer) else {
            debugPrint("unable to get image from sample buffer")
            return
        }
```

```
        print("did receive image frame")
        // process image here
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        let guesture = UIPinchGestureRecognizer(target:self, action:#selector(pinchAction))
        self.view.addGestureRecognizer(guesture)
        self.pinchAction(guesture)
        self.addCameraInput()
        self.addPreviewLayer()
        self.addVideoOutput()
        self.captureSession.startRunning()
    }

    @IBAction func pinchAction(_ sender: UIPinchGestureRecognizer) {
        let captureDevice = AVCaptureDevice.default(.builtInWideAngleCamera, for: .video, position: .back)!
        let device = captureDevice
        var zoom = device.videoZoomFactor * sender.scale
        sender.scale = 1.0
        var error:NSError!
        do{
            try device.lockForConfiguration()
            defer {device.unlockForConfiguration()}
            if zoom >= device.minAvailableVideoZoomFactor && zoom <= device.maxAvailableVideoZoomFactor {
                device.videoZoomFactor = zoom
            }else{
                NSLog("Unable to set videoZoom: (max %f, asked %f)", device.activeFormat.videoMaxZoomFactor, zoom);
            }
        }catch error as NSError{
            NSLog("Unable to set videoZoom: %@", error.localizedDescription);
        }catch _{
        }
    }

}
```

Global Health Microsurgery Training With Cell Phones

Jason Wang - Sep 10, 2020, 10:52 PM CDT

Title: Global Health Microsurgery Training With Cell Phones

Date: 09/10/2020

Content by: Jason

Present:

Goals:

Content:

https://pubmed.ncbi.nlm.nih.gov/32294075/

- Background
  -
- Challenge
  - Global surgery teaching/training limited by resources available
  - Surgical loupes and operating microscopes used to perform microsurgery are expensive
  - Identify low-cost alternatives to microsurgery suturing
- Innovation
  - To use cell phone camera with zoom capacity to teach and practice microsurgery suturing.
- Results
  - Cell phones with camera feature is widely available in low-/middle- income countries
  - Protocol
    - Cell phone placed on stand over microsurgery practice station
    - Camera used to zoom and focus on the suturing station to mimic a surgical field
    - 9 attending surgeons and 7 residents practiced microsurgery under the magnification of the cell phone camera
  - The Stanford Microsurgery and Resident Training Scale was used to track progress.
  - Feedback survey given to participants
- Conclusions
  -

Conclusions/action items:

Can Smartphones Be Used to Perform Video-Assisted Microanastomosis? An Experimental Study

Jason Wang - Sep 11, 2020, 12:12 PM CDT

Title: Can Smartphones Be Used to Perform Video-Assisted Microanastomosis? An Experimental Study

Date: 09/10/2020

Content by: Jason

Present:

Goals:

Content:

https://journals-sagepub-com.ezproxy.library.wisc.edu/doi/pdf/10.1177/1553350618822626

- Introduction
    - The cost of microsurgical microscopes are a limiting factor
    - Best results obtained when using cameras with similar or higher resolution than microsurgical microscopes
    - Advantages
        - Better ergonomy
        - Autofocus system
        - Deployment of the operating room personnel
    - Disadvantages
        - The lack of stereoscopic view
        - The presence of several cables next to the surgical field
        - The limitation of the surgical field by the required support
        - The need to acquire specific cameras.
    - Several studies have shown that smartphones have use in laparoscopic box trainers and histopathological evaluation
    - Study aimed to evaluate the use of the smartphones' magnification system for video-assisted microanastomosis in rats
- Methods
    - 15 male Wistar rats
        - 200 - 300 g
        - 12 - 15 weeks old
    - Divided rats into 3 groups of 5
    - 3 groups
        - Microscope Group (MG)
            - DFVasconcelos microsurgery microscope
        - iPhone Group (IG)
            - iPhone 7
        - Samsung Group (SG)
            - Samsung Galaxy S7
    - Smartphones kept at 12cm (IG) or 10cm (SG) height
    - The phones were connected to a 55-inch, high-definition television set
    - The image magnification was 35× for both smartphone groups (2.5× optical and 14× digital zoom) and 40× for MG.
- Results
    - Both smartphone groups were unable to perform arterial anastomosis or neurorrhaphy
        - Due to insufficient image quality
- Discussion
    - The rectangular operative field display (ie, television) was very helpful and more comfortable than the limited circular field seen under the microscope, which sends direct light into the eyes, causing headache and nausea after long periods of use.
    - Wi-Fi technology integrated in smartphones avoids the use of cables, simplifying the procedure
        - Allows for image sharing, which can be used for streaming video transmission of surgeries for educational purposes
        - If internet connection is poor, it can limit the usefulness of these devices
    - can be used as a magnification system for training both at a macroscopic scale and in microsurgery
        - may reduce initial cost of microsurgical training
- Conclusion

Conclusions/action items:

Protocol

Jason Wang - Sep 21, 2020, 6:10 PM CDT

Title: Protocol

Date: 09/14/2020

Content by: Jason

Present:

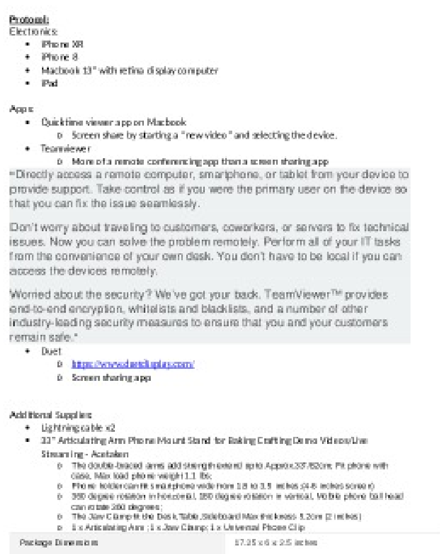
Goals:

- Materials
  - iPhone XR
  - iPhone 8
  - Macbook 13" w/ retina display computer
- Apps
  - Quicktime viewer
  - Teamviewer
  - Duet
- Other Supplies
  - Lightning cable
  - 33" articulating arm phone mount stand
  - Google cardboard

Content:

Conclusions/action items:

Jason Wang - Sep 21, 2020, 5:43 PM CDT



Protocol:

Electronics:

- iPhone XR
- iPhone 8
- Macbook 13" with retina display computer
- iPad

Apps:

- Quicktime viewer app on Macbook
  - Screen share by starting a "new video" and selecting the device.
- Teamviewer
  - More of a remote conferencing app than a screen sharing app

"Directly access a remote computer, smartphone, or tablet from your device to provide support. Take control as if you were the primary user on the device so that you can fix the issue seamlessly.

Don't worry about traveling to customers, coworkers, or servers to fix technical issues. Now you can solve the problem remotely. Perform all of your IT tasks from the convenience of your own desk. You don't have to be local if you can access the devices remotely.

Worried about the security? We've got your back. TeamViewer™ provides end-to-end encryption, whitelists and blacklists, and a number of other industry-leading security measures to ensure that you and your customers remain safe."

- Duet
  - https://www.duetdisplay.com/
  - Screen sharing app

Additional Supplies:

- Lightning cable x2
- 33" Articulating Arm Phone Mount Stand for Baking Crafting Demo Videos/Live Streaming - Acetaken
  - The double-braced arms add strength extend up to Approx.33"/82cm; Fit phone with case. Max load phone weight 1.1 lbs;
  - Phone holder can fit smartphone wide from 1.8 to 3.5 inches (4-6 inches screen)
  - 360 degree rotation in horizontal, 180 degree rotation in vertical. Mobile phone ball head can rotate 360 degrees;
  - The Jaw Clamp fit the Desk,Table,Sideboard Max thickness 5.2cm (2 inches)
  - 1 x Articulating Arm ; 1 x Jaw Clamp; 1 x Universal Phone Clip

Package Dimensions	17.25 x 6 x 2.5 inches

Protocol.docx(16.9 KB) - download

MM51 Brochure

Jason Wang - Sep 21, 2020, 6:30 PM CDT

Title: MM51 Brochure

Date: 09/21/2020

Content by: Jason

Present:

Goals:

- Larger lens gives better resolution
- 8:1 zoom system
- Clear operation on vessels of less than 1 mm
- 4k camera and monitor

Content:

Conclusions/action items:

Jason Wang - Sep 21, 2020, 6:21 PM CDT

MM51_Brochure_Web_6.2019.pdf(840.7 KB) - download

Jason Wang - Sep 21, 2020, 6:51 PM CDT

MM51 YOH Surgical Microscope System

User's Manual
Ver. 1.0

Mitaka Kohki Co., Ltd.

August 2014

US-MM51-Manual-141223lr2.pdf(668.5 KB) - download

Orbeye

Jason Wang - Sep 21, 2020, 6:51 PM CDT

Title: Orbeye

Date: 09/21/2020

Content by: Jason

Present:

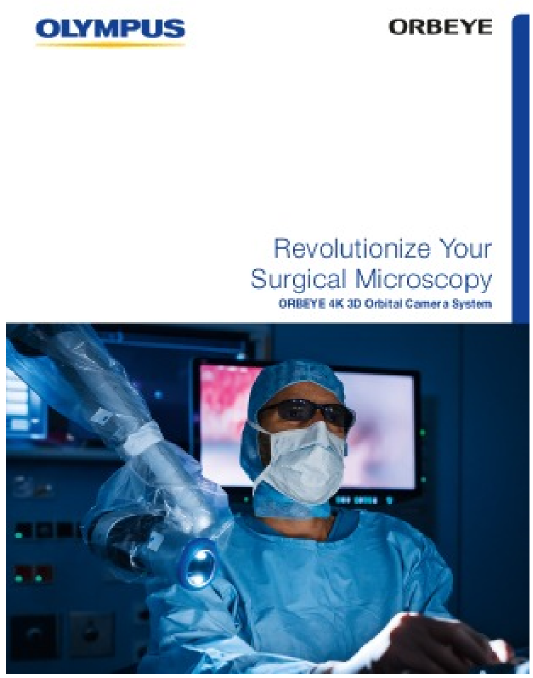
Goals:

- 4k, 3D visualization
  - no image latency
  - 26x magnification
  - 4k 3D imaging chain
- Bright Observation Modes
  - Cold-light LEDs with virtually no heat generation
  - IR - Infrared
  - BL - Blue light
  - NBI - Narrow band imaging
- Ergonomic Benefits
  - Heads-up posture
  - ample working space
-

Content:

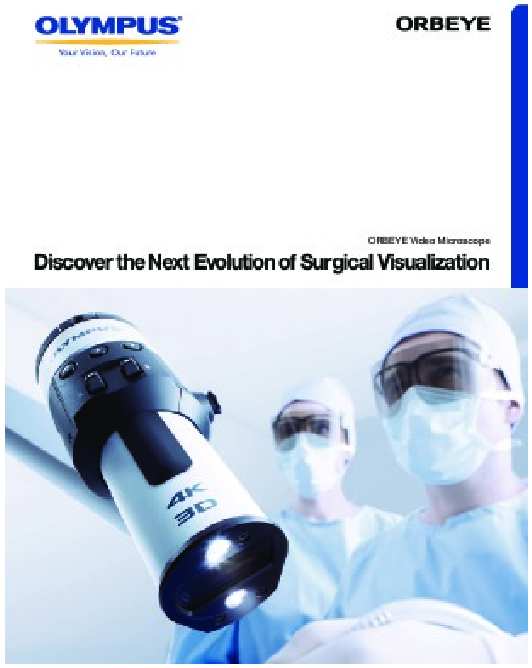
Conclusions/action items:

Jason Wang - Sep 21, 2020, 6:31 PM CDT



ORBEYE_concept_brochure_53297.pdf(7.8 MB) - download

Jason Wang - Sep 21, 2020, 6:31 PM CDT



N8600837_ORBEYE_0914_REV092717.pdf(1.5 MB) - download

3D Glasses

Jason Wang - Oct 01, 2020, 11:07 PM CDT

Title: 3D Glasses

Date:

Content by: Jason

Present:

Goals:

- How does 3D work?
    - Uses 2 slightly different images to trick the brain into thinking it's seeing a 3D image
- Anaglyphic image
    - A stereoscopic motion or still picture in which the right component of a composite image usually red in color is superposed on the left component in a contrasting color to produce a three-dimensional effect when viewed through correspondingly colored filters in the form of spectacles.
    - Uses the natural physiological distance between the eyes , 2 inches, to create the effect of 2 slightly different viewpoints.
        - binocular vision
    - the colored filters separate the two different images so each image only enters one eye. Your brain puts the two pictures back together.
- Active 3D Shutter Glasses
    - an HDTV will display one image to your left eye and one image to your right eye.
    - The effective frame-rate is halved
    - need a screen of at least 120 Hz
- Passive (Polarized) Glasses
    - The display shows two overlapping images and the glasses have polarized lenses
    - Each lens is polarized so that it can see only one of the two overlapping images.

Content:

Conclusions/action items:

Google Cardboard

Jason Wang - Sep 11, 2020, 12:41 PM CDT

Title: Google Cardboard

Date: 09/11/2020

Content by: Jason

Present:

Goals:

Content:

Conclusions/action items:

Green Shop Pass

Jason Wang - Mar 08, 2019, 7:25 PM CST

Title: Green Shop Pass

Date: 03/08/2019

Content by: Me

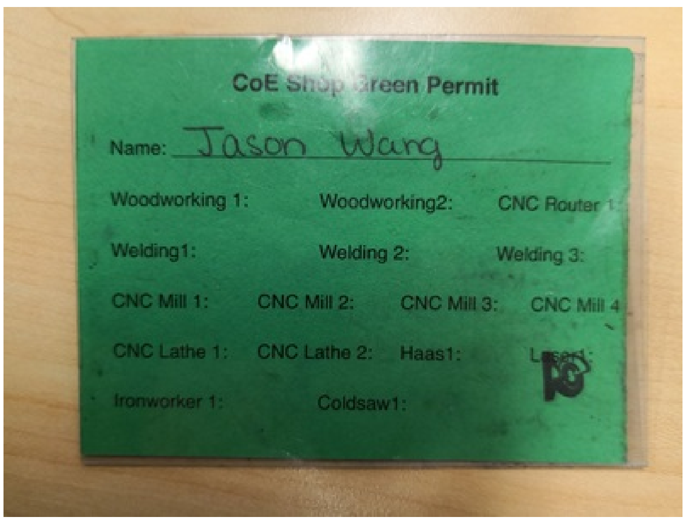
Present: Me

Goals:

Content:

Conclusions/action items:

Jason Wang - Mar 08, 2019, 7:29 PM CST



IMG_20190308_192752.jpg(2.5 MB) - download

Jason Wang - Mar 08, 2019, 7:29 PM CST

IMG_20190308_192817.jpg(2.6 MB) - download

09/11/2020 - Advisor Meeting

Jason Wang - Sep 12, 2020, 10:54 AM CDT

Title: Advisor Meeting

Date: 09/11/2020

Content by: Jason

Present: All

Goals:

Content:

- Look into
  - ipad pro LIDAR
  - App production
    - xcode and SwiftUI
  - Similar devices
- Make sure to get a lot of useful information from the client to know where to begin research and work on the project

Conclusions/action items:

09/18/2020 - Advisor Meeting

Jason Wang - Oct 07, 2020, 3:39 PM CDT

Title: Advisor Meeting

Date: 09/18/2020

Content by: Jason

Present: All

Goals:

Content:

- Talked about possible research pathways to follow
- Xiaoxuan mentioned neural networks
- Overall, team agreed to focus on researching VR glasses and 3D depth perception technologies

Conclusions/action items:

09/25/2020 - Advisor Meeting

Jason Wang - Oct 07, 2020, 3:40 PM CDT

Title: Advisor Meeting

Date: 09/25/2020

Content by: Jason

Present: All

Goals:

Content:

- Discussed the Design matrix ideas
    - Advisor concerned over posture of the microsurgeon while using the monitor from design 1
    - Discussed the use of 3d glasses and the different types
    - Discussed the presentation guidelines

Conclusions/action items:

10/09/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:33 PM CST

Title: Advisor Meeting

Date: 10/09/2020

Content by: Jason

Present: All

Goals:

Content:

- Talked about the deliverables
- Began to work on setting up our coding environment to write the apple App

Conclusions/action items:

10/16/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:35 PM CST

Title: Advisor Meeting

Date: 10/16/2020

Content by: Jason

Present: All

Goals:

Content:

- Talked about how the team has set up a Github page for collaboration
- Talked about the start of the image conversion to anaglyph through Matlab

Conclusions/action items:

10/23/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:37 PM CST

Title: Advisor Meeting

Date: 10/23/2020

Content by: Jason

Present: All

Goals:

Content:

- No really big update, other than troubleshooting the Github and the visual studio working environment

Conclusions/action items:

10/30/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:38 PM CST

Title: Advisor Meeting

Date: 10/30/2020

Content by: Jason

Present: All

Goals:

Content:

- Discussed the show and tell poster
- Went over the progress made on the app development

Conclusions/action items:

11/06/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:39 PM CST

Title: Advisor Meeting

Date: 11/06/2020

Content by: Jason

Present: All

Goals:

Content:

- Updated advisor on the completion of the camera function of the app
    - Discussed the need to produce the magnification function
- Discussed slow, but steady progress on the image conversion software

Conclusions/action items:

11/13/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:41 PM CST

Title: Advisor Meeting

Date: 11/13/2020

Content by: Jason

Present: All

Goals:

Content:

- Same content as the previous Advisor meeting
- Still working on developing magnification software
- Still working on image converter

Conclusions/action items:

11/20/2020 - Advisor Meeting

Jason Wang - Dec 09, 2020, 1:42 PM CST

Title: Advisor Meeting

Date: 11/20/2020

Content by: Jason

Present: All

Goals:

Content:

- Updated advisor on the completion of the magnification
    - Additionally, discussed next step of data output analysis
- Demonstrated completed anaglyph image conversion

Conclusions/action items:

09/11/2020 - Client Meeting

Jason Wang - Sep 12, 2020, 11:54 AM CDT

Title: Client Meeting

Date: 09/11/2020

Content by: Jason

Present: All

Goals:

Content:

- Client
    - Ellen - 2nd year resident
    - Dr. Poore
    - + a couple other not present
- iPhone
    - Has good resolution and magnification
    - Limitation through lack of depth perception
        - Possible Solution: VR Goggles
- Microsurgical microscopes are usually used, but expensive
    - Alternatives:
        - Exoscope
            - High resolution camera that projects to a monitor
        - Possibly iPhone/smartphone setup
- Model 1
    - iPhone + iPhone holder/stand + the basic camera app
        - ~$20-30 not including iPhone
    - iPhone connected to Macbook Pro
        - Using a lightning cable attached to both
    - Use quicktime player to do a live screenshare of the iPhone
    - Advantages
        - Magnification maintained
        - Easy to set up
        - Minimal delay between actual phone camera and the screenshare
        - Relatively inexpensive
    - Disadvantages
        - Unable to use the full screen of the macbook???
        - iPhone needs to be vertical for proper image orientation?
        - Need for an external light source
        - Lack of Depth Perception
    - Ideal Set up for end-product
- Model 2
    - iPhone + Reality A app?
    - Advantages
        - Allows you to change interpupillary distance
        - Provides a full screen image
        - Able to use the iPhone's built in light
        - Capable of recording
    - Disadvantages
        - No zoom/magnification capability
        - Apparently not user friendly
            - Not touch sensitive or Bad touch programming
- Model 3
    - iPhone + iPhone + Google Cardboard + Teamviewer/Duet
    - Advantages
        -
    - Disadvantages
        - Significant delay between actual camera and screenshare (.5-1s)
        - Decreased user friendliness

        - Multiple devices and connections
- Magnification of microscope anywhere from 2x to 25x
- Prefer more direct view, peripherals are not as important
- Prefer to try and use smartphone, but not necessary

Conclusions/action items:

09/18/2020 - Client Meeting

Jason Wang - Oct 07, 2020, 3:43 PM CDT

Title: Client Meeting

Date: 09/11/2020

Content by: Jason

Present: All

Goals:

Content:

- Discussed the design ideas that could be used for the project itself
- Asked clarifying question from the week prior
- Client
  - expressed interesting in purchasing 3d lens attachment for the iPhone
  - Acknowledged that the project doesn't need to follow the smartphone route, but would be prefered
  - Questioned if the google cardboard lenses could be modified for a longer focal length

Conclusions/action items:

09/29/2020 - Client Meeting

Jason Wang - Oct 07, 2020, 3:46 PM CDT

Title: Client Meeting

Date: 09/29/2020

Content by: Jason

Present: All

Goals:

Content:

- Followed up about the changing of the google cardboard lenses to increase focal length
    - Does not seem feasible as the size of the google cardboard would have to be increased
- Posture
    - Posture does not seem to be an issue as stated by the clients
    - Having a good quality image is more important
    - Prior to the surgery
        - Everything is prepped to the surgeons preferences
            - Camera zoom
            - hand position
            - resolution
            - camera position
- Main concern
    - Eliminate as many screens as possible
        - reduce the delay/latency between screens
-

Conclusions/action items:

10/09/2020 - Client Meeting

Jason Wang - Dec 09, 2020, 11:50 AM CST

Title: Client Meeting

Date: 10/09/2020

Content by: Jason

Present: All

Goals:

Content:

- Decided to meet on a biweekly basis
- Introduced final design plan to the clients
    - the application creation and the needs it will fulfil
- Clarified the zoom needs
    - 2-5x
    - 15-25x

Conclusions/action items:

10/23/2020 - Client Meeting

Jason Wang - Dec 09, 2020, 12:06 PM CST

Title: Client Meeting

Date: 10/23/2020

Content by: Jason

Present: All

Goals:

Content:

- Updated the clients on the progress of the applications
    - The work on creating the camera function
    - Creation of the anaglyph image conversion
- Asked about the testing of the stereoscopic iPhone attachment
    - No real update at the moment

Conclusions/action items:

11/06/2020 - Client Meeting

Jason Wang - Dec 09, 2020, 12:21 PM CST

Title: Client Meeting

Date: 11/06/2020

Content by: Jason

Present: All

Goals:

Content:

- Updated the clients on the progress of the applications
  - The completion of the camera function of the application
  - Further work needed on the magnification function of the application
  - Some progress made on the anaglyph conversion in matlab code
- Again asked about the testing of the stereoscopic iPhone attachment
  - No real update at the moment

Conclusions/action items:

11/20/2020 - Client Meeting

Jason Wang - Dec 09, 2020, 12:24 PM CST

Title: Client Meeting

Date: 11/20/2020

Content by: Jason

Present: All

Goals:

Content:

- Updated the clients on the progress of the applications
    - The completion of the magnification function of the application
    - Further work needed on the analysis of the data stream from the application to be used for the anaglyph conversion
    - Working prototype anaglyph conversion that is working, however the conversion is very slow and there is a major loss in resolution
- And again asked about the testing of the stereoscopic iPhone attachment
    - Plan to do testing in the next few weeks

Conclusions/action items:

12/04/2020 - Client Meeting

Jason Wang - Dec 09, 2020, 12:37 PM CST

Title: Client Meeting

Date: 12/04/2020

Content by: Jason

Present: All

Goals:

Content:

- Gave the final update on the current progress of the application
- Recieved update on the completion of the 3d attachment testing
    - Not viable
        - the focal point of the lens is too great, leading to about .5 meter long
        - Fish-eyed
        - Cause a drastic increase in microsuturing time
- Talked about when the next semester of the project will continue

Conclusions/action items:

09/11/2020 - Team Meeting

Jason Wang - Sep 12, 2020, 10:54 PM CDT

Title: Team Meeting

Date: 09/12/2020

Content by: Jason

Present: All

Goals:

Content:

- Went over the first client meeting and the main takeaways that we have gotten
- Set up a weekly team meeting time and responsibilities
- Split up research responsibilities
    - Jason + Xiaoxuan
        - VR apps and screensharing techniques
    - Jiong + Martin
        - Improving connection/streaming speed

Conclusions/action items:

2020/9/17 GoPro Cameras

Martin Janiszewski - Sep 17, 2020, 1:02 PM CDT

Title: GoPro Cameras Used in Innovative Microsurgery 3D Video Training Project

Date: 9/17/2020

Content by: Martin Janiszewski

Present: NA

Goals: To discover other cameras that could be used in microsurgery

Content:

- Currently, most microscopic surgical procedures are recorded from one microscope eyepiece, producing videos in 2D format and therefore lacking critical depth information.
- 3D imagery allows observers to visualize the surgery as if they were the actual surgeon or review their own cases with a faculty member after performing the surgery.

How:

- The technique electronically links two GoPro cameras so that video capture by each camera is perfectly synchronized. The cameras are then mounted to either the left or right microscope eyepiece to record simultaneous videos during the surgery.

Conclusions/action items:

3D/depth perception could be achieved using GoPro cameras as hardware and with the appropriate software.

Source: https://www.uabmedicine.org/-/gopro-cameras-used-in-innovative-microsurgery-3d-video-training-project

2020/9/17 Lightning Cable Transfer Speeds

Martin Janiszewski - Sep 17, 2020, 2:19 PM CDT

Title: Lightning Cable Transfer Speeds

Date: 9/17/2020

Content by: Martin Janiszewski

Present: NA

Goals: To determine why the client experiences considerable lag when using lightning cables

Content:

- Lightning cables transfer data at USB 2.0 speed, which is 480Mbps/60MBps
- USB-C can handle USB 3.0 speed, with transfer speeds as fast as 5Gbps/640MBps (USB 3.1 Gen 1), or 10Gbps (USB 3.1 Gen2).
- Some iPads have a Lightning connector supporting USB 3.0
- No one really knows exactly what speed Lightning can handle as Apple doesn't release all specifications. In fact, an update on the standards of Lightning cables has not been seen in over eight years.

Conclusions/action items:

The issue with the Lightning cable may be that its data transfer rate is too slow to quickly transfer the high quality video capture of the camera being used.

Of course, there may be lag generated by the streaming device itself, which needs to be researched separately.

Sources:

https://www.macrumors.com/2015/11/12/ipad-pro-lightning-port-usb-3-0-speeds/

https://www.ipitaka.com/blogs/news/usb-c-vs-lightning-which-is-the-future

2020/9/25 IEEE Automatic Real-time 2D-to-3D Conversion for scenic views

Martin Janiszewski - Sep 25, 2020, 1:30 PM CDT

Title: Real-time 2D-to-3D Conversion

Date: 9/25/2020

Content by: Martin Janiszewski

Present: NA

Goals: To determine if software exists that can create real-time 2D to 3D video footage

Content:

Using machine-learning algorithms, multiple depth factors (monocular depth cues) can be utilized to estimate the depth map that would be used for 2D to 3D conversion. The problem is this sort of processing is done in post and not in real-time. Researchers at the University of British Columbia, Vancouver have developed a process that generates high quality depth maps for scenic views in real-time.

Their process involves analyzing three depth cues: haze, vertical edges, and sharpness to estimate a sparse depth map. They then obtain the full depth map from the sparse map using an edge-aware interpolation method. This creates a high quality depth map by forming feature vectors from each 4x4 pixel block.

Haze is an atmospheric diffusion of light that results in contrast loss effect in images. This is most apparent in outdoor and distant images.

For haze extraction,

$$I_{dark}(x,y) = \min_{k\in(r,g,b)}\left(\min_{(x,y)\in\Omega} I_k(x,y)\right)$$

where omega denotes the 4x4 pixel block and Ik(x,y) represents the minimum values for each spatial location in the red, green, and blue channels within the pixel block.

For the same vertical coordinate, objects with larger vertical edges appear closer compared to the objects with smaller (or without) vertical edges.

For vertical edge extraction, a 2D convolution with a 4x4 mask is defined as:

$$\bar{v}(x,y) = \sum_{(x,y)\in\Omega}\frac{v(x,y)}{M}$$

where omega represents the 4x4 block, M is the number of pixels in that block, and v(x,y) denotes the horizontal gradient value.

For sharpness extraction, a second derivative of a Gaussian filter to measure object blurriness is used.

Depth map generation:

To obtain the full depth map, the sparse depth map is propagated to the entire frame:

$$(L + \lambda D_s)\alpha = \lambda s,$$

where alpha denotes the full depth map, s is the sparse map, lambda is a scalar that balances the sharpness of the sparse depth map and the smoothness of the full depth map, Ds represents a diagonal matrix whose diagonal element is 1 for sparse pixels and 0 otherwise, and L is a matting Laplacian matrix whose (i,j)th element is:

$$\sum_{k|(i,j)\in\omega_k}\left[\delta_{ij} - \frac{1}{|\omega_k|}\left(1 + (I_i - \mu_k)\left(\sum_k + \frac{\epsilon}{|\omega_k|}I_3\right)^{-1}(I_j - \mu_k)\right)\right]$$

where Ii and Ij are the colors of the input image. I3 is a 3x3 identity matrix, delta(ij) is the Kronecker delta, sigma(k') is a covariance matrix, mu(k) is a mean of the colors in window omega(k) of size 4x4, and absolutevalue(omega(k)) is the number of pixels in this window.

To solve the optimization problem above and find the full depth map, the depth map to the global optimum of the following cost function is:

$$J(\alpha) = \alpha^T L\alpha + \lambda(\alpha - s)(\alpha - s)$$

Functionally allowing the globally optimal estimate for the full depth map to be based on the sparse depth map.

Conclusions/action items:

This research was done for the purpose of creating 2D to 3D *scenic* views, but their criteria for depth cues could be of use in the application we want to use - high quality magnification for microsurgery. This involves creating a software that can immediately analyze video footage and do the specific calculations to create a depth map.

Martin Janiszewski - Sep 25, 2020, 1:30 PM CDT


Automatic Real-time 2D-to-3D Conversion for scenic views

IEEE_Automatic_Real-time_2D-to-3D_Conversion_for_scenic_views.pdf(1.1 MB) - download

2020/10/1 Heads-up 3D Microscopy: An Ergonomic and Educational Approach to Microsurgery

Martin Janiszewski - Oct 07, 2020, 11:18 AM CDT

Title: Heads-up 3D Microscopy

Date: 10/1/2020

Content by: Martin Janiszewski

Present: NA

Goals: To explore experimental approaches to 3D Microscopy

Content:

The purpose of this study was to evaluate the feasibility of performing heads-up 3D microscopy as a more ergonomic alternative to traditional microsurgery.

A TrueVision Systems product was tested for this experiment, allowing a surgeon to work while keeping their head up, something our client wants to achieve.

Table 1. Results of Questionnaire Completed by 8 Microsurgeons Rating Heads-up 3D Microscopy on a 1 to 10 Scale Relative to Traditional Binocular Microscopy (1 = Vastly Inferior to Traditional Microscopy, 5 = Equivalent, 10 = Vastly Superior)

Microsurgeon	Age (y)	Image Resolution	Depth Perception	Field of View	Neck/Upper Back Comfort	Lower Back Comfort	Technical Feasibility	Educational Value
Veterinary surgeon 1	51	9	9	5	8	8	6	10
Veterinary surgeon 2	61	5	5	5	9	9	6	10
Fellow 1	30	3	3	5	5	5	6	10
Fellow 2	31	5	5	3	5	5	6	10
Fellow 3	33	5	5	5	8	8	5	10
Junior attending	35	5	5	5	8	8	5	10
Senior attending 1	50	5	3	3	5	5	3	10
Senior attending 2	54	3	3	5	7	7	3	10
Average 1–10 rating compared with traditional microscopy		5.0	5.0	4.5	6.9	6.9	5.0	10.0

Measures that were compared included image quality, technical difficulty, neck/back comfort, and educational value. Fellow: plastic surgery independent fellow. Jr. attending: attending plastic surgeon with <20 y of microsurgical experience. Sr. attending: attending plastic surgeon with >20 y of microsurgical experience.

6 of the 8 participants (75%) rated neck and back comfort to be superior with heads-up 3D microscopy compared with traditional microsurgery (both with average rating of 6.9 of 10), whereas the remaining 2 participants rated comfort to be equivalent

With regard to image resolution, field of view, and technical feasibility, 75% of participants found heads-up microscopy to be equivalent or superior to traditional microsurgery

In 63% of participants, depth perception was found to be equivalent or superior with heads-up 3D microscopy

In addition, 100% of participants found heads-up 3D microscopy to be a more valuable educational and interactive experience - projecting to a large screen for multiple viewers is an intended goal for a client

It is worth noting that "neck and back postures used in traditional binocular microsurgery can expose surgeons to higher biomechanical joint loads and an increased risk of work-related injuries. In 2010, Capone et al reported >75% of plastic surgeons having work-related neck and back symptoms, with a 3-fold higher likelihood in surgeons performing >3 hours per week of microsurgery. In the same study, symptomatic surgeons were 5 times more likely to develop impairment or disability during their career."

- Hence why a heads up approach

Conclusions/action items:

This study helps purport that heads-up 3D microscopy is a preferable method to traditional microscopy and is a great teaching tool, which is useful for the training that our client's device is made for.

While this article does not provide any information to assist in developing our own system, it supports that our device has value to microsurgeons (in case the support of our client and their lab of microsurgeons wasn't enough incentive).

Martin Janiszewski - Oct 07, 2020, 11:21 AM CDT

OPEN

IDEAS AND INNOVATIONS

Special Topic

Heads-up 3D Microscopy: An Ergonomic and Educational Approach to Microsurgery

Bernardino M. Mendez, MD
Michael V. Chiodo, MD
Darl Vandevender, MD, FACS
Parit A. Patel, MD

Summary: Traditional microsurgery can lead surgeons to use postures that cause musculoskeletal fatigue, leaving them more prone to work-related injuries. A new technology from TrueVision transmits the microscopic image onto a 3-dimensional (3D) monitor, allowing surgeons to operate while sitting/standing in a heads-up position. The purpose of this study was to evaluate the feasibility of performing heads-up 3D microscopy as a more ergonomic alternative to traditional microsurgery. A feasibility study was conducted comparing heads-up 3D microscopy and traditional microscopy by performing femoral artery anastomoses on 8 Sprague-Dawley rats. Operative times and patency rates for each technology were compared. The 8 microsurgeons completed a questionnaire comparing image quality, comfort, technical feasibility, and educational value of the 2 technologies. Rat femoral artery anastomoses were successfully carried out by all 8 microsurgeons with each technology. There was no significant difference in anastomosis time between heads-up 3D and traditional microscopy (average times, 34.5 and 35.8 minutes, respectively; $P = 0.68$). Heads-up 3D microscopy was rated superior in neck and back comfort by 75% of participants. Image resolution, field of view, and technical feasibility were found to be superior or equivalent in 75% of participants, whereas 63% evaluated depth perception to be superior or equivalent. Heads-up 3D microscopy is a new technology that improves comfort for the microsurgeon without compromising image quality or technical feasibility. Its use has become prevalent in the field of ophthalmology and may also have utility in plastic and reconstructive surgery. *(Plast Reconstr Surg Glob Open 2016;4:e717; doi: 10.1097/GOX.0000000000000727; Published online 25 May 2016.)*

Over the past 2 decades, new innovations have led to a dramatic increase in the prevalence of microsurgery. Despite these advancements, traditional binocular microsurgery can lead surgeons to use neck and back postures that cause musculoskeletal fatigue and injuries.[1-3] As a result, chronic poor posture and musculoskeletal strain can lead to increased rates of disability and reduced surgical longevity.[1-4]

In 2010, TrueVision Systems (Santa Barbara, Calif.) developed a new 3-dimensional (3D) technology that allows the microsurgeon to sit or stand in a more physiologic "heads-up" posture while operating. We have described this as "heads-up 3D microscopy." This system transmits the surgical mi-

From the Division of Plastic and Reconstructive Surgery, Department of Surgery, Loyola University Chicago, Health Sciences Campus, Stritch School of Medicine, Maywood, Ill.

Presented orally at the annual meeting of the American Society of Plastic Surgeons, October 10–14, 2014, Chicago, Ill.

Received for publication July 10, 2014; accepted March 23, 2016.

DOI: 10.1097/GOX.0000000000000727

Disclosure: *The authors have no financial interest to declare in relation to the content of this article. The Research and Graduate Programs at Loyola University Medical Center, Stritch School of Medicine funded the animal laboratory equipment and rats used in this study. The Article Processing Charge was paid for by the authors.*

Supplemental digital content is available for this article. Clickable URL citations appear in the text.

Heads-up_3D_Microscopy_An_Ergonomic_and_Educationa.pdf(513.5 KB) - download

2020/12/6 - anaglyph3d script

Martin Janiszewski - Dec 09, 2020, 7:31 AM CST

Title: anaglyph3d script

Date: 12/7/2020

Content by: Martin Janiszewski

Present: NA

Goals: To research how to convert an image into anaglyph in Matlab

Content:

Matlab has a function called "stereoAnaglyph" which can convert a pair of camera images into anaglyph, however, more control over the creation of the anaglyph image is required for our purposes.

https://www.mathworks.com/help/vision/ref/stereoanaglyph.html

Since the team does not have much experience with image and video editing in general, it was reasonable to search on the internet for any example code people have posted involving converting images to anaglyph.

One such program was called "anaglyph3d": https://www.mathworks.com/matlabcentral/fileexchange/59537-anaglyph3d

This script is quite advanced as it uses Matlab UI components for more fine tuning of image editing, which is computation intensive and not necessary for the creation of a basic anaglyph script.

The script is analyzed section by section as follows:

```
function anaglyph3d(image1, image2, filter1, filter2)

% Saving function inputs to workspace.
assignin('base','image1',image1);
assignin('base','image2',image2);
assignin('base','fil1',filter1);
assignin('base','fil2',filter2);

% Reading the images.
img1 = imread(image1);
img2 = imread(image2);
```

The script is a function and has input parameters that must be established. The script needs to identify and read image inputs and detect filter settings (allowing for a customizable anaglyph filter).

```
% Adjusting the colors in both images, based on input filters.
switch filter1
    case 'red'
        img1(:,:, 2:3) = 0;
    case 'green'
        img1(:,:, 1:2:3) = 0;
    case 'blue'
        img1(:,:, 1:2) = 0;
    case 'cyan'
        img1(:,:, 1) = 0;
    case 'magenta'
        img1(:,:, 2) = 0;
end
switch filter2
    case 'red'
        img2(:,:, 2:3) = 0;
    case 'green'
        img2(:,:, 1:2:3) = 0;
    case 'blue'
        img2(:,:, 1:2) = 0;
    case 'cyan'
        img2(:,:, 1) = 0;
    case 'magenta'
        img2(:,:, 2) = 0;
end
```

The script includes various options for anaglyph filters for either image that would be used in the conversion. Inputs corresponding to these colors will shade the images so that they have these colors.

```
% Rotating the image, based on value in "rotation" variable
img2 = imrotate(img2,rotation,'bilinear','crop');

% Shifting images when values of both horizontal and vertical shifts are positive.
if horizontalShift > -1 && verticalShift > -1
    resultImg1 = img1(1:end-verticalShift, (1+horizontalShift):end, :);
    resultImg2 = img2((1+verticalShift):end, 1:end-horizontalShift, :);

    % Shifting images when value of horizontal shift is negative.
elseif horizontalShift < 0 && verticalShift > -1
    resultImg1 = img1(1:end-verticalShift, 1:end+horizontalShift, :);
    resultImg2 = img2(1+verticalShift:end, (1-horizontalShift):end, :);

    % Shifting images when value of vertical shift is negative.
elseif horizontalShift > -1 && verticalShift < 0
    resultImg1 = img1((1-verticalShift):end, 1+horizontalShift:end, :);
    resultImg2 = img2(1:end+verticalShift, 1:end-horizontalShift, :);

    % Shifting images when values of both horizontal and vertical shifts are negative.
elseif horizontalShift < 0 && verticalShift < 0
    resultImg1 = img1((1-verticalShift):end, 1:end+horizontalShift, :);
    resultImg2 = img2(1:end+verticalShift, (1-horizontalShift):end, :);
end
```

The script includes an advanced feature to rotate and shift either image directly using Matlab, but this feature will be overlooked due to its complexity.

```
% Our first image is just a sum of input images with adjusted colors.
resultImage = img1+img2;
```

Once the images have been filtered and modified in Matlab as needed, a final image is created.

In the script there are additional functions written that support the special UI that the script provides, however, Matlab will not be the final program to run the code as the code is intended to run in Swift with Xcode for Apple devices, and it cannot be expected for these functions to work in the final environment. Additionally, all of these additional functions are intensive and all processing power needs to be minimized to ensure that the code can process images and video in real-time.

Conclusions/action items:

With the understanding of how a properly written image anaglyph conversion script works, it will be easier to develop a similar and simpler script for the team's purposes.

Martin Janiszewski - Dec 07, 2020, 3:11 AM CST

```
function anaglyph3d(image1, image2, filter1, filter2)
%       -- Anaglyph3D - Create stereoscopic 3D anaglyph images --
%
%   Anaglyph3D takes two pictures as an input (one shot from the left
%   and one shot from the right) and merges them into one 3D stereoscopic
%   image. The resulting 3D anaglyph stereo image can be adjusted
%   by shifting and/or rotating one of the input images.
%   After such adjustement, the result can be saved to a JPG image file.
%
%   Function Syntax:
%         anaglyph3d(image1, image2, filter1, filter2)
%
%   Examples (work with sample pictures included in the package):
%         anaglyph3d('MadameBoudonL.jpg','MadameBoudonR.jpg','red','cyan')
%         anaglyph3d('VoltaireL.jpg','VoltaireR.jpg','red','blue')
%         anaglyph3d('PascalL.jpg','PascalR.jpg','green','magenta')
%
%   - 'image1' is static, it doesn't move at all,
%   - 'image2' is the one that is shifted or rotated,
%   - image files must be located in the same folder as this file,
%   - both images must have the same resolution (size in pixels),
%   - large (high resolution) images are not recommended due to overload;
%   - supported anaglyph filters for left and right eyes are:
%     red, green, blue, cyan and magenta.
%
%   The following keys are used for adjustments and saving:
%
%   arrowkey "up"      - shifts 'image2' upwards,
%   arrowkey "down"    - shifts 'image2' downwards,
%   arrowkey "left"    - shifts 'image2' to the left,
%   arrowkey "right"   - shifts 'image2' to the right,
%   "+"(plus) or "a"   - rotates 'image2' anti-clockwise,
%   "-"(minus) or "c"  - rotates 'image2' clockwise,
%   "space"            - swaps the filters (e.g. "red-blue" to "blue-red"),
%   "s"                - saves the stereo anaglyph image to 'output.jpg',
%   "h"                - opens help window.
%
%   Anaglyph3D has been developed and tested on MATLAB R2012a and R2015a.
%   -----------------------------------------------------------------------
%   I am grateful to Professor Igor Podlubny (http://www.tuke.sk/podlubny/)
%   for the formulating this exercise and for providing the sample photos
%   (6 files), which are included with this file.
%
%   Author: Matej Mikulszky, October 2016
%   email: matej.mikulszky@student.tuke.sk
%   Student, Technical University, Kosice, Slovakia
%
%   For updates, please check the following link:
%   http://bit.ly/mikulszky_mathworks
%   -----------------------------------------------------------------------
%
%   UPDATES:
%       Version 1.1: April 6th, 2017
%         - Fixed width or height on startup, for all types of screens.
%         - Application window is now resizable.
%         - Anaglyph's aspect ratio is preserved when shifting/rotating image.
%         - Using more "fluid" function for drawing anaglyph.

%% Basic function - Creates stereoscopic image and waits for key press.
global resultImage M Position ignore
ignore = 1;

% Saving function inputs to workspace.
```

anaglyph3d.m(13.5 KB) - download

2020/12/6 - Anaglyph from One Image

Martin Janiszewski - Dec 09, 2020, 7:53 AM CST

Title: Anaglyph from One Image

Date: 12/6/2020

Content by: Martin Janiszewski

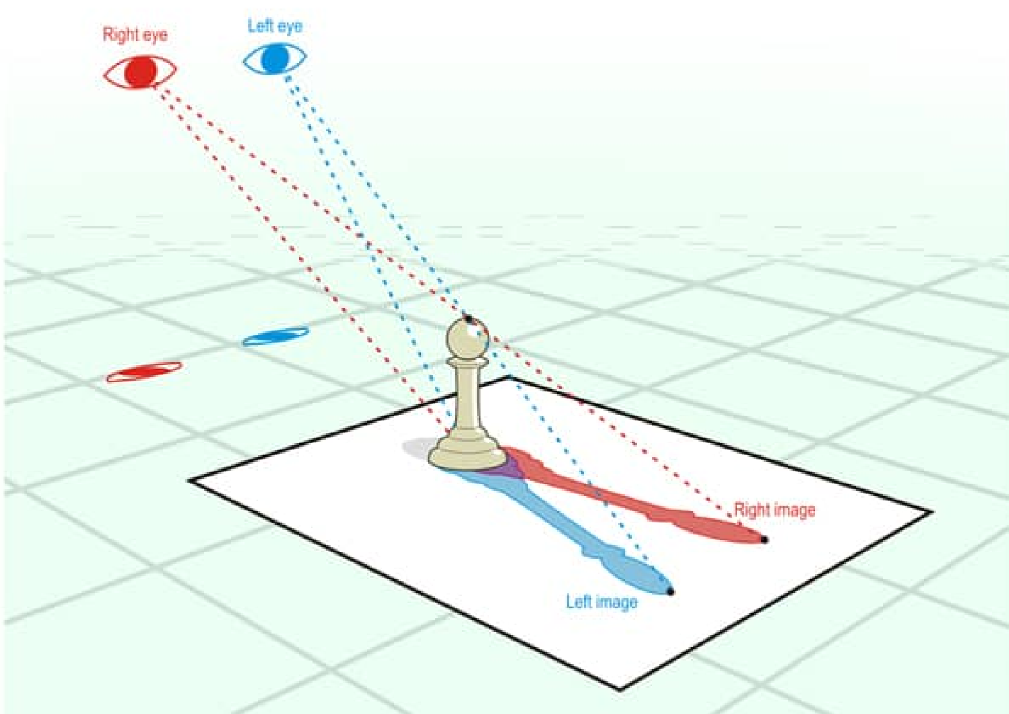
Present: NA

Goals: To learn to develop a script that can turn a single image into anaglyph

Content:

With a simple camera lens and no additional camera hardware, only a single image is available for anaglyph development.

VR as a whole can be viewed in one of two ways: monoscopically or stereoscopically - https://immersionvr.co.uk/blog/monoscopic-vs-stereoscopic-360-vr/



Stereoscopic VR requires two viewpoints to generate depth perception. Two images are seen, but one from each eye - the brain puts these images together as one image and a sense of depth perception is achieved. Anaglyph image creation also makes used of stereoscopic images (two different images that are different viewpoints of the same scene) to create a single image that provides depth perception using 3D glasses.

Monoscopic VR uses a single viewpoint/image to generate depth perception. Images are merged together to develop a single image with depth cues. Anaglyph using a single viewpoint makes use of merged version of the same image modified to create an anaglyph effect that can be seen as depth perception using 3D glasses.

However, to know how to use a single image to create an anaglyph image, one must know the Matlab commands to modify an image to look different. Here are the commands to be learned:

Imread is necessary to read images to process them in matlab:

https://www.mathworks.com/help/matlab/ref/imread.html

```
A = imread(filename)
```

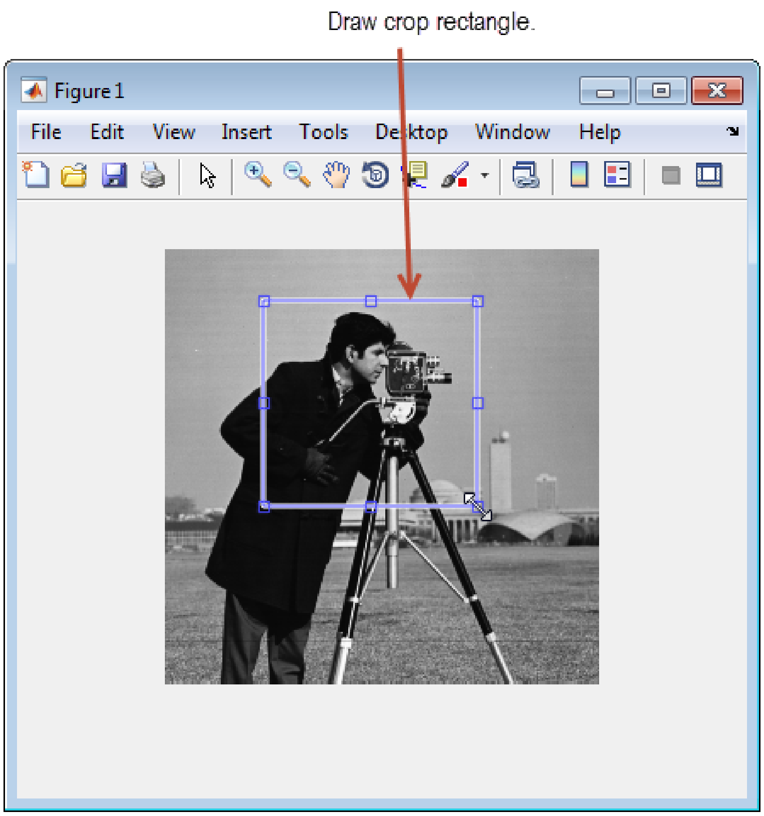
Imrotate is necessary to rotate images to create an anaglyph effect from one image by using two identical images rotated in opposite directions:

https://www.mathworks.com/help/images/ref/imrotate.html

```
J = imrotate(I,angle)
```

Imcrop feature to crop images to focus on properly filtered sections of images:

https://www.mathworks.com/help/images/ref/imcrop.html




```
[J,rect] = imcrop(I);
```

```
I2 = imcrop(I,[75 68 130 112]);
```

Matlab supports an easy means to crop images by hand, and this information can be saved as a matrix. For each possible image resolution IE 720p, 1080p, etc, this function needs to be used, but since the iPhone camera resolution is known, this only needs to be done once in Matlab and the values for the crop can be saved in the code for use outside of Matlab.

Imshow makes it easy to view cropped images during testing:

https://www.mathworks.com/help/matlab/ref/imshow.html

```
imshow(I)
```

```
imshow(corn_gray)
```

Imwrite allows for these images to be saved as a new anaglyph image:

https://www.mathworks.com/help/matlab/ref/imwrite.html

```
imwrite(A,filename)
```

Conclusions/action items:

With these functions learned, it is now possible to write a script that can be called as a function in the process to convert a single image into an anaglyph image.

Later on, assignin will be used to read frames instead of images, as MATLAB sees frames from a video differently than it views images.

https://www.mathworks.com/help/matlab/ref/assignin.html#d122e50167

```
assignin(ws,var,val)
```

2020/12/7 - Converting Video into Frames for Analysis

Martin Janiszewski - Dec 09, 2020, 8:08 AM CST

Title: Converting Video into Frames for Analysis

Date: 12/7/2020

Content by: Martin Janiszewski

Present: NA

Goals: To learn how to process a video in Matlab and edit it frame by frame

Content:

For the most part, a video is a series of images played within a certain time interval very much like an animation. Cameras save frames at a certain speed and then play all the images at the speed to generate a video.

In order to convert a video into anaglyph, each of these frames must be processed with the anaglyph effect, and put back together into a video to have anaglyph video. This is known as a time intensive process as high quality video at a high framerate takes a longer time to process all the images in a second.

Matlab can be used to process videos, but the means to do this must be learned. The internet provides example code for taking a video, processing each frame with a specific effect, and saving all of the frames back into a new video.

The script processed here is referred to as ExtractMovieFrames: https://www.mathworks.com/matlabcentral/answers/58111-how-to-convert-video-in-to-image-frames-using-matlab

The basic functionality of the code is explained as follows:

```
% Open the rhino.avi demo movie that ships with MATLAB.
% First get the folder that it lives in.
folder = fileparts(which('Globe.avi')); % Determine where demo folder is (works with all versions).
% Pick one of the two demo movies shipped with the Image Processing Toolbox.
% Comment out the other one.
movieFullFileName = fullfile(folder, 'Globe.avi');
% movieFullFileName = fullfile(folder, 'traffic.avi');
% Check to see that it exists.
if ~exist(movieFullFileName, 'file')
    strErrorMessage = sprintf('File not found:\n%s\nYou can choose a new one, or cancel', movieFullFileName);
    response = questdlg(strErrorMessage, 'File not found', 'OK - choose a new movie.', 'Cancel', 'OK - choose a new
    if strcmpi(response, 'OK - choose a new movie.')
        [baseFileName, folderName, FilterIndex] = uigetfile('*.avi');
        if ~isequal(baseFileName, 0)
            movieFullFileName = fullfile(folderName, baseFileName);
        else
            return;
        end
    else
        return;
    end
end
```

The code looks for a .avi video file to read, and provides a basic UI that provides an error in case it fails to read a movie.

```
videoObject = VideoReader(movieFullFileName)
% Determine how many frames there are.
numberOfFrames = videoObject.NumberOfFrames;
vidHeight = videoObject.Height;
vidWidth = videoObject.Width;

numberOfFramesWritten = 0;
% Prepare a figure to show the images in the upper half of the screen.
figure;
%   screenSize = get(0, 'ScreenSize');
% Enlarge figure to full screen.
set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

% Ask user if they want to write the individual frames out to disk.
promptMessage = sprintf('Do you want to save the individual frames out to individual disk files?');
button = questdlg(promptMessage, 'Save individual frames?', 'Yes', 'No', 'Yes');
if strcmp(button, 'Yes')
    writeToDisk = true;
```

The script then records some information about the videos characteristics and displays the video and its new effects as each frame is processed. The script also provides example code to program a means to save the processed video as a new file, if the user chooses to do so.

```
for frame = 1 : numberOfFrames
    % Extract the frame from the movie structure.
    thisFrame = read(videoObject, frame);

    % Display it
    hImage = subplot(2, 2, 1);
    image(thisFrame);
    caption = sprintf('Frame %4d of %d.', frame, numberOfFrames);
    title(caption, 'FontSize', fontSize);
    drawnow; % Force it to refresh the window.
```

The code then processes each frame of the movie one by one, and displays it so the user can see how video is being processed. At this point code for processing each frame can be implemented. However, any additional features besides processing and immediately displaying the new video footage will not be useful for the sake of the project as any additional functions are processor intensive and will impede real-time video conversion.

```
% Alert user that we're done.
if writeToDisk
    finishedMessage = sprintf('Done!  It wrote %d frames to folder\n"%s"', numberOfFramesWritten, outputFolder);
else
    finishedMessage = sprintf('Done!  It processed %d frames of\n"%s"', numberOfFramesWritten, movieFullFileName)
end
disp(finishedMessage); % Write to command window.
uiwait(msgbox(finishedMessage)); % Also pop up a message box.
```

Once the code finishes processing the video, it will provide a user with a message that it is completed. This will not be required as the program required by the client will display video footage immediately.

It could be possible to record the video footage, however, it would need to process the video footage in parallel as it might not be useful for microsurgeons to watch video footage post-operation with an anaglyph filter. Thus, normal video footage would be recorded, but this would require powerful hardware to be able to show anaglyph video and save normal video all in real-time, which might not be feasible.

Conclusions/action items:

Using this example code, a basis for video anaglyph conversion can be developed.

Martin Janiszewski - Dec 07, 2020, 3:11 AM CST

```
% Demo to extract frames and get frame means from a movie and save individual
frames to separate image files.
% Then rebuilds a new movie by recalling the saved images from disk.
% Also computes the mean gray value of the color channels
% And detects the difference between a frame and the previous frame.
% Illustrates the use of the VideoReader and VideoWriter classes.
% A Mathworks demo (different than mine) is located here
http://www.mathworks.com/help/matlab/examples/convert-between-image-sequences-
and-video.html

clc;    % Clear the command window.
close all;  % Close all figures (except those of imtool.)
imtool close all;  % Close all imtool figures.
clear;  % Erase all existing variables.
workspace;  % Make sure the workspace panel is showing.
fontSize = 22;

% Open the rhino.avi demo movie that ships with MATLAB.
% First get the folder that it lives in.
folder = fileparts(which('Globe.avi')); % Determine where demo folder is (works
with all versions).
% Pick one of the two demo movies shipped with the Image Processing Toolbox.
% Comment out the other one.
movieFullFileName = fullfile(folder, 'Globe.avi');
% movieFullFileName = fullfile(folder, 'traffic.avi');
% Check to see that it exists.
if ~exist(movieFullFileName, 'file')
	strErrorMessage = sprintf('File not found:\n%s\nYou can choose a new one,
or cancel', movieFullFileName);
	response = questdlg(strErrorMessage, 'File not found', 'OK - choose a new
movie.', 'Cancel', 'OK - choose a new movie.');
	if strcmpi(response, 'OK - choose a new movie.')
		[baseFileName, folderName, FilterIndex] = uigetfile('*.avi');
		if ~isequal(baseFileName, 0)
			movieFullFileName = fullfile(folderName, baseFileName);
		else
			return;
		end
	else
		return;
	end
end

try
	videoObject = VideoReader(movieFullFileName)
	% Determine how many frames there are.
	numberOfFrames = videoObject.NumberOfFrames;
	vidHeight = videoObject.Height;
	vidWidth = videoObject.Width;

	numberOfFramesWritten = 0;
	% Prepare a figure to show the images in the upper half of the screen.
	figure;
	%	screenSize = get(0, 'ScreenSize');
	% Enlarge figure to full screen.
	set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

	% Ask user if they want to write the individual frames out to disk.
	promptMessage = sprintf('Do you want to save the individual frames out to
individual disk files?');
	button = questdlg(promptMessage, 'Save individual frames?', 'Yes', 'No',
'Yes');
	if strcmp(button, 'Yes')
```

ExtractMovieFrames.m(9.6 KB) - download

2020/12/6 - Basic Anaglyph Photo Conversion

Martin Janiszewski - Dec 09, 2020, 8:40 AM CST

Title: Martin Janiszewski

Date: 2020/12/6

Content by: Martin Janiszewski

Present: NA

Goals: To develop a program that can convert an image into an anaglyph image

Content:

Since MATLAB was being used, despite most of the team's prior experience with the program being around signal analysis and statistics/calculations, programs on the Internet could be used as a reference as to how input data images must be processed.

Refer to the research on "anaglyph3d" for the full analysis of how this script works.

```
function [resultImage] = anaglyph(image1, image2, filter1, filter2)
%% Basic function - Creates stereoscopic image and waits for key press.

% Saving function inputs to workspace.
assignin('base','image1',image1);
assignin('base','image2',image2);
assignin('base','fil1',filter1);
assignin('base','fil2',filter2);

% Reading the images.
img1 = imread(image1);
img2 = imread(image2);

% Adjusting the colors in both images, based on input filters.
switch filter1
    case 'red'
        img1(:,:, 2:3) = 0;
    case 'green'
        img1(:,:, 1:2:3) = 0;
    case 'blue'
        img1(:,:, 1:2) = 0;
    case 'cyan'
        img1(:,:, 1) = 0;
    case 'magenta'
        img1(:,:, 2) = 0;
end
switch filter2
    case 'red'
        img2(:,:, 2:3) = 0;
    case 'green'
        img2(:,:, 1:2:3) = 0;
    case 'blue'
        img2(:,:, 1:2) = 0;
    case 'cyan'
        img2(:,:, 1) = 0;
    case 'magenta'
        img2(:,:, 2) = 0;
end

resultImage = img1 + img2;
```

This is the basis of my anaglyph photo conversion program. Assuming an input of two transposed (and not completely identical) images, they are overlaid on top of each other and either image can be given one of 5 colors to be adjusted to.

Here is the first rough usage of this program:

Here, the quality is bad since two images with a phone camera from different angles were taken without standardizing how the images were taken.

Nevertheless, the left and right images can be seen and it can be noticed that one is cyan colored and the other is red. With anaglyph 3D glasses, some effect would be seen, although it would look most likely look bad.

Conclusions/action items:

How exactly the two images are seen relative to each other must be standardized. Without a 3D camera attachment, the two images must be generated digitally as there is only camera view from a typical iPhone. A script for that must be made as well as a script for the ability to process videos into anaglyph.

Martin Janiszewski - Dec 07, 2020, 1:21 AM CST

```
function [resultImage] = anaglyph(image1, image2, filter1, filter2)
%% Basic function - Creates stereoscopic image and waits for key press.

% Saving function inputs to workspace.
assignin('base','image1',image1);
assignin('base','image2',image2);
assignin('base','fil1',filter1);
assignin('base','fil2',filter2);

% Reading the images.
img1 = imread(image1);
img2 = imread(image2);

% Adjusting the colors in both images, based on input filters.
switch filter1
    case 'red'
        img1(:,:, 2:3) = 0;
    case 'green'
        img1(:,:, 1:2:3) = 0;
    case 'blue'
        img1(:,:, 1:2) = 0;
    case 'cyan'
        img1(:,:, 1) = 0;
    case 'magenta'
        img1(:,:, 2) = 0;
end
switch filter2
    case 'red'
        img2(:,:, 2:3) = 0;
    case 'green'
        img2(:,:, 1:2:3) = 0;
    case 'blue'
        img2(:,:, 1:2) = 0;
    case 'cyan'
        img2(:,:, 1) = 0;
    case 'magenta'
        img2(:,:, 2) = 0;
end

resultImage = img1 + img2;
```

anaglyph.m(993 Bytes) - download

2020/12/6 - one_image_anaglyph script

Martin Janiszewski - Dec 09, 2020, 8:40 AM CST

Title: one_image_anaglyph script

Date: 12/6/2020

Content by: Martin Janiszewski

Present: NA

Goals: To be able to generate anaglyph images from one image

Content:

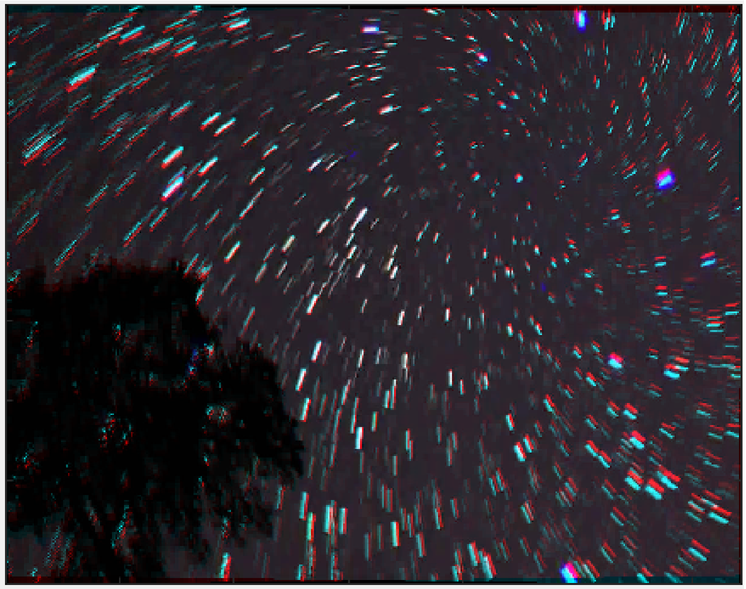
Refer to Anaglyph From One Image for the discussion on monoscopic vs. stereoscopic VR, and for more information as to the program was written.

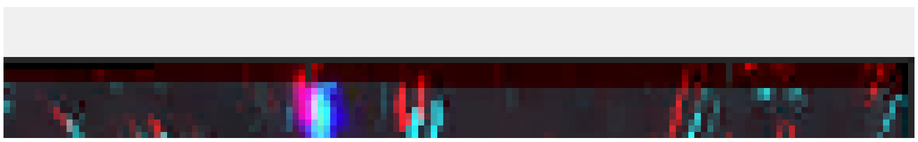
```
function [x] = one_image_anaglyph(image)

assignin('base','frame',image);

parallaxAngle = 0.5 ;       % You can change this to suit yourself.

%leftImage = imread(image);
leftImage = image;

J = imrotate(leftImage,(parallaxAngle));

% rightImage = imread(image);
rightImage = image;

K = imrotate(rightImage, (-parallaxAngle));

%Here's how this works:
% [J1,rect] = imcrop(J) activates a custom crop menu where I must crop the
% image J, and it will be saved as J1. This must be exclusive per image.
% The parameters for the crop window are saved in rect.
% We can apply imcrop with rect as below with K1

rect = [9.510000000000000,21.510000000000000,1.057980000000000e+03,1.883980000000000e+03];
 %rect = [100, 200, 400 300];
J1 = imcrop(J, rect);
K1 = imcrop(K, rect);

%subplot(1,2,1), imshow(J1);
%subplot(1,2,2), imshow(K1);

imwrite(J1, 'leftImage.jpg');
imwrite(K1, 'rightImage.jpg');

% x1 = 'leftImage.jpg';
% y1 = 'rightImage.jpg';

[x] = anaglyph('leftImage.jpg', 'rightImage.jpg', 'red', 'cyan');
```

This is the latest modified version of the script, with some notes depending on how it is being used. As it currently is, the script reads frames, which will bed explained in the video anaglyph conversion page.

This script was first tested using a single image. Using an adjustable angle, the image is duplicated and given two versions, one that is slightly rotated to the left, and one that is rotated to the same magnitude in the other direction.

This, however, presents an issue when the images overlap each other, as seen here:





At the corners of an uncropped pair of images will be a region where the images won't overlap due to the fact that the images are angled at 5 degrees in opposite images.

To overcome this, the images are cropped to display only the region that is overlapping. This reduces the image size by a small amount, but it is mostly negligible if the image quality is not compromised.

For example, a 434x342 image can be reduced to 421x332 to cut out the edges of the pictures that stick out. This is about a 5.83% reduction in image size, which ultimately isn't a lot. This might vary based on the proportions of the image's width to length, but in practicality it is not a significant issue.

The crop is also specific for each image so it needs to be redone when using an image of a different size. The Matlab imcrop feature makes this rather easy, and a phone camera will have a set image resolution that won't need to be established more than once.

Once the images are cropped, they are used as the inputs for the anaglyph function and an anaglyph image is produced from a single image.

Here is a better example:

Conclusions/action items:

Now that a single image can be turned into anaglyph, the final step in this process is to develop a script that can process a video into images. This leads to further complications, but for now the first thing to do is to develop something that works.

Martin Janiszewski - Dec 07, 2020, 1:20 AM CST

```
function [s] = one_image_anaglyph(image)

assignin('base','frame',image);

parallaxAngle = 0.5 ;        % You can change this to suit yourself.

%leftImage = imread(image);
leftImage = image;

J = imrotate(leftImage,(parallaxAngle));

% rightImage = imread(image);
rightImage = image;

K = imrotate(rightImage, (-parallaxAngle));

%Here's how this works:
% [J1,rect] = imcrop(J) activates a custom crop menu where I must crop the
% image J, and it will be saved as J1. This must be exclusive per image.
% The parameters for the crop window are saved in rect.
% We can apply imcrop with rect as below with K1

%rect =
[9.510000000000000,21.510000000000000,1.057900000000000e+03,1.051900000000000e+0
3];
 rect = [100, 200, 400 300];
J1 = imcrop(J, rect);
K1 = imcrop(K, rect);

%subplot(1,2,1), imshow(J1);
%subplot(1,2,2), imshow(K1);

imwrite(J1, 'leftImage.jpg');
imwrite(K1, 'rightImage.jpg');

% x1 = 'leftImage.jpg';
% y1 = 'rightImage.jpg';

[s] = anaglyph('leftImage.jpg', 'rightImage.jpg', 'red', 'cyan');
```

one_image_anaglyph.m(1 KB) - download

2020/12/7 - ExtractMovieFrames with Anaglyph Conversion Full Script

Martin Janiszewski - Dec 09, 2020, 8:45 AM CST

Title: ExtractMovieFrames with Anaglyph Conversion Full Script

Date: 12/7/2020

Content by: Martin Janiszewski

Present: NA

Goals: To develop a script that can convert video into images for anaglyph conversion, and then back into video.

Content:

Since this script is based off of the one mentioned in "Converting Video into Image...", refer to it for more detail on how this version of the script works.

The original script was made as an example of one form of video analysis, and so far it has been used with a modified version of it to prove that an anaglyph video can be made. The parts that are currently being used will be focused on here:

1.) Read Video

```
% First get the folder that it lives in.
folder = fileparts(which('star_trails.avi')); % Determine where demo folder is (works with all versions).
% Pick one of the two demo movies shipped with the Image Processing Toolbox.
% Comment out the other one.
movieFullFileName = fullfile(folder, 'star_trails.avi');
```

For the moment, this script only reads .avi files while iPhone videos are saved in .mov. It is possible to integrate usage of .mov files, but for now, if need be, a different program can convert .mov to .avi. The concern here is any drop in quality that comes with converting a video file.

2.) Record Information on Video

```
videoObject = VideoReader(movieFullFileName);
% Determine how many frames there are.
numberOfFrames = videoObject.NumFrames;
vidHeight = videoObject.Height;
vidWidth = videoObject.Width;
```

This program will read videos into frames. What this means is that if the video plays at 30 fps, the program will read 30 frames for every second of video. Inherently it takes more power to process higher resolution images, as well as video that is recorded at a higher framerate.

3.) Process the Frames into Anaglyph

```
for frame = 1 : numberOfFrames
    % Extract the frame from the movie structure.
    thisFrame = read(videoObject, frame);

    % Display it
    %hImage = subplot(1, 1, 1);
    %image(thisFrame);
    %caption = sprintf('Frame %4d of %d.', frame, numberOfFrames);
    %title(caption, 'FontSize', fontSize);
    %drawnow; % Force it to refresh the window.

    % Anaglyph it and display it
     aImage = subplot(1,1,1);


     image(one_image_anaglyph(thisFrame));

    % [x] = one_image_anaglyph(thisFrame);
    % image(anaglyph(x, y, 'red', 'cyan'));


    %caption = sprintf('Frame %4d of %d.', frame, numberOfFrames);
    %title(caption, 'FontSize', fontSize);
    drawnow; % Force it to refresh the window.
```

For very little active code and a lot of commented out code, this part took a while to figure out. In combination with modifying the other two scripts to work as functions, the fact that a frame is NOT an image for the purposes of processing an image needed to be accommodated for.

This part of the code takes all the frames of the video and process them 1 by 1 and displays them as fast as the code can process it. At the base level, this is what the code should do for the user - display anaglyph video as soon as the camera creates the video feed. Potentially the code should also record non-anaglyph video for analysis of footage at a later time, but for now the code cannot optimally display anaglyph video fast enough.

Since the code is currently processing saved footage and not active footage, a difference which in general should hopefully be easy to accommodate for, it will display additional information about the footage once the program is done:

```
General Properties:
         Name: 'Jiong.avi'
         Path: 'C:\Users\User1\Documents\MATLAB'
     Duration: 2.1595
  CurrentTime: 0
    NumFrames: 63

Video Properties:
        Width: 1080
       Height: 1920
    FrameRate: 29.9700
 BitsPerPixel: 24
  VideoFormat: 'RGB24'
```

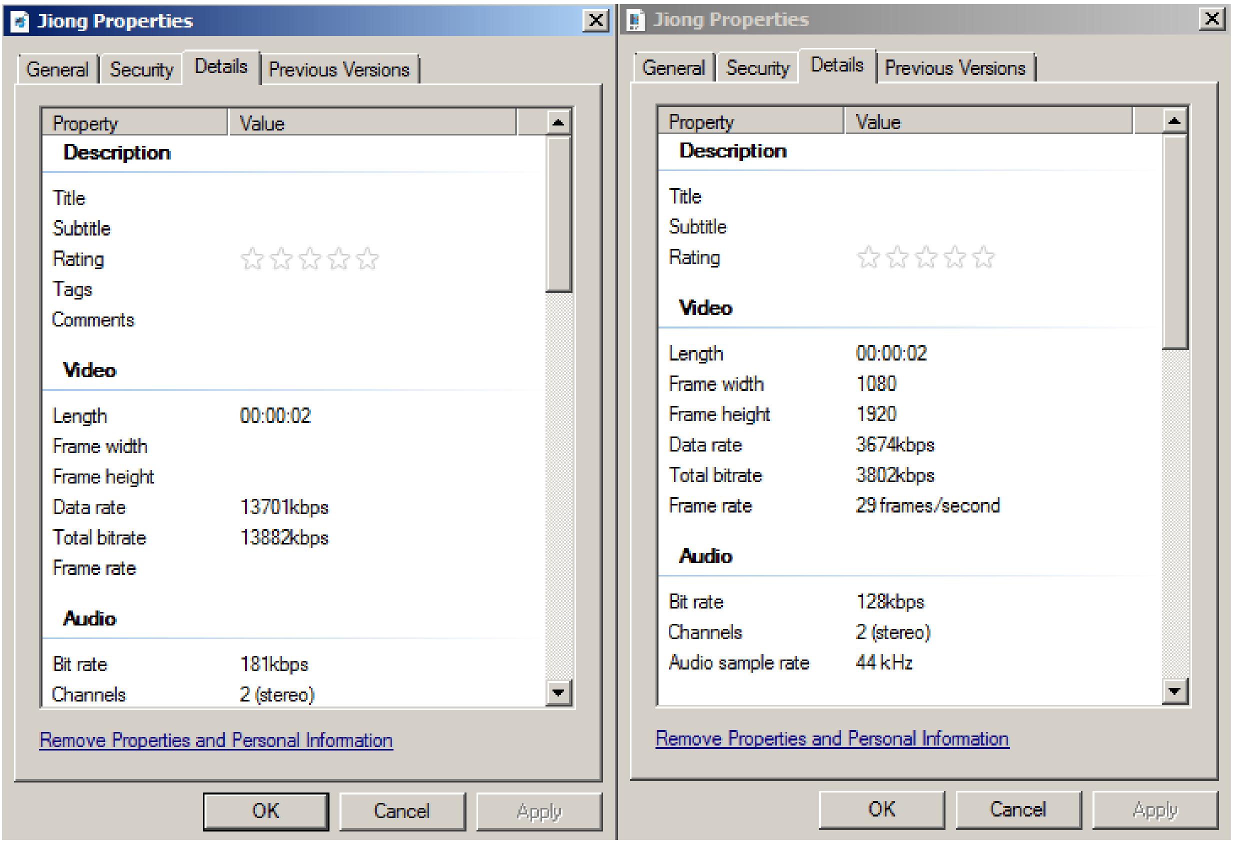
```
General Properties:
         Name: 'Rabbit.avi'
         Path: 'C:\Users\Misiek\Documents\MATLAB'
     Duration: 125.9607
  CurrentTime: 0
    NumFrames: <Calculating...> learn more

Video Properties:
        Width: 320
       Height: 240
    FrameRate: 15.0000
 BitsPerPixel: 24
  VideoFormat: 'RGB24'
```

Before this comparison is explained, there is a bug where Numframes fails to be expressed even though it is easily recorded, the NumFrames for the second table here is 1889.

On the left is a more higher quality video, however, it is processed at a rate of ~2.4 frames/second whereas the lower quality video on the right is played close to its native resolution of 15 fps.

This code is constantly being improved, but at some point hardware limitations come into play and will affect the feasibility of the project.

Video Conversion Notes:



On the left here the .mov or QuickTime version of the sample video that is originally saved by the iPhone can be seen, and on the right is the .avi Video Clip that has been converted from it, in which the computer is capable of recognizing its properties. The iPhone 12 from which this footage originates from boasts the ability to record in 4k resolution at 60 fps. If this sort of quality can be processed in real-time for our client, it would be amazing, but due to a lack of such technology on the market, real-time anaglyph video is too niche a demand or this form of video processing in real-time is very hardware demanding. In any case, once it becomes .avi, the computer sees it as a 1080x1920 footage at ~30 fps, which may seem a bit low, but it is still too much for the current program with this hardware (3.4GHz processor, 16gB ram) to run in real-time.

Conclusions/action items:

This is as far as the code goes. Many things still need to be worked on - this entire program needs to run on Swift/Xcode to work on Apple devices for the client's purposes, which has many hurdles which will be explained in the report for this fall semester. This part of the code needs to be rewritten anew so that it is as straightforward to run as possible to better optimize the code to run better. There is also the issue of hardware testing and performance which will be mentioned in the report.

The most important issue is to get the program to run in real-time at the highest resolution and framerate as code and hardware will allow.

Martin Janiszewski - Dec 07, 2020, 1:21 AM CST

```
% Demo to extract frames and get frame means from a movie and save individual
frames to separate image files.
% Then rebuilds a new movie by recalling the saved images from disk.
% Also computes the mean gray value of the color channels
% And detects the difference between a frame and the previous frame.
% Illustrates the use of the VideoReader and VideoWriter classes.
% A Mathworks demo (different than mine) is located here
http://www.mathworks.com/help/matlab/examples/convert-between-image-sequences-
and-video.html

clc;    % Clear the command window.
close all;  % Close all figures (except those of imtool.)
imtool close all;  % Close all imtool figures.
clear;  % Erase all existing variables.
workspace;  % Make sure the workspace panel is showing.
fontSize = 22;

% Open the rhino.avi demo movie that ships with MATLAB.
% First get the folder that it lives in.
folder = fileparts(which('Globe.avi')); % Determine where demo folder is (works
with all versions).
% Pick one of the two demo movies shipped with the Image Processing Toolbox.
% Comment out the other one.
movieFullFileName = fullfile(folder, 'Globe.avi');
% movieFullFileName = fullfile(folder, 'traffic.avi');
% Check to see that it exists.
if ~exist(movieFullFileName, 'file')
	strErrorMessage = sprintf('File not found:\n%s\nYou can choose a new one,
or cancel', movieFullFileName);
	response = questdlg(strErrorMessage, 'File not found', 'OK - choose a new
movie.', 'Cancel', 'OK - choose a new movie.');
	if strcmpi(response, 'OK - choose a new movie.')
		[baseFileName, folderName, FilterIndex] = uigetfile('*.avi');
		if ~isequal(baseFileName, 0)
			movieFullFileName = fullfile(folderName, baseFileName);
		else
			return;
		end
	else
		return;
	end
end

try
	videoObject = VideoReader(movieFullFileName)
	% Determine how many frames there are.
	numberOfFrames = videoObject.NumberOfFrames;
	vidHeight = videoObject.Height;
	vidWidth = videoObject.Width;

	numberOfFramesWritten = 0;
	% Prepare a figure to show the images in the upper half of the screen.
	figure;
	%	screenSize = get(0, 'ScreenSize');
	% Enlarge figure to full screen.
	set(gcf, 'units','normalized','outerposition',[0 0 1 1]);

	% Ask user if they want to write the individual frames out to disk.
	promptMessage = sprintf('Do you want to save the individual frames out to
individual disk files?');
	button = questdlg(promptMessage, 'Save individual frames?', 'Yes', 'No',
'Yes');
	if strcmp(button, 'Yes')
```

ExtractMovieFrames_with_Anaglyph_Conversion.m(9.6 KB) - download

Martin Janiszewski - Dec 07, 2020, 3:11 AM CST

Target_Video_Processing_slower_speed.mkv(919.1 KB) - download

Martin Janiszewski - Dec 07, 2020, 3:11 AM CST

Sample_Simpler_Video_faster_speed.mkv(2.6 MB) - download

08/29/2020 background research

Xiaoxuan Ren - Sep 28, 2020, 2:35 PM CDT

Title: Background Research

Date: 08/29/2020

Content by: Xiaoxuan Ren

Goals: Get an overall understanding of the project

Content:

generation of 3D images
from left and right view

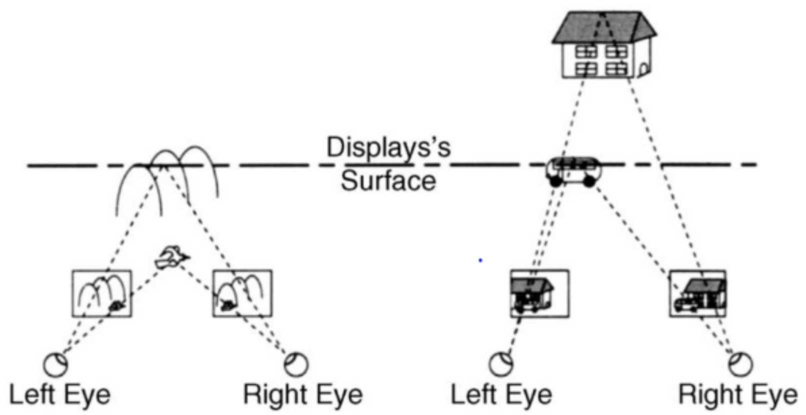


Fig.2. Determination of the left and right eye images from a 2D object moving to the left.

Cite from paper: A Real Time 2D to 3D Image Conversion Techniques

Current Algorithms

- Depth Image-Based Rendering (DIBR)
  - estimate a depth map from the left view
  - render the right view by DIBR
- Deep Learning Neural Networks
  - take single-view 2D images and their depth maps as supervision
  - learn a mapping from 2D image to depth map

Possible Solution

treat view synthesis as image reconstruction

offline spatial transformer module &

The offline module pre-processes a color+depth image database to speed up the subsequent depth estimation.

online depth-inferring module

The online module infers a depth prior from a color query image using the previous database as training data

---

Xiaoxuan Ren - Sep 28, 2020, 2:33 PM CDT

BME_iPhone_VR_20200911_1_.pptx(233.4 KB) - download

09/11/2020 real-time 3D rendering

Xiaoxuan Ren - Sep 28, 2020, 4:09 PM CDT

Title: real-time 3D rendering

Date: 09/11/2020

Content by: Xiaoxuan Ren

Goals: background knowledge about real-time 3D video

Content:

Definition:

3D rendering is the process of producing an image based on three-dimensional data stored on your computer.

With 3D rendering, your computer graphics converts 3D wireframe models into 2D images with 3D photorealistic.

Processing Time:

Rendering can take from seconds to even days for a single image or frame.

Two Major Types:

- real-time
- offline or pre-rendering

Minimum Speed:

24 frames/sec. (The main goal is to achieve the highest possible degree of photorealism at an acceptable minimum rendering speed which is usually 24 frames/sec. That's the minimum a human eye needs in order to create the illusion of movement.)

Because in real-time rendering (most common in video games or interactive graphics) the 3D images are calculated at a very high speed so that it looks like the scenes, which consist of multitudes of images, occur in real time when players interact with your game.

That's why interactivity and speed play important roles in the real-time rendering process. For example, if we want to move a character in our scene, we need to make sure that the character's movement is updated before drawing the next frame, so that it's displayed at the speed with which the human eye can perceive as natural movement.

Conclusions/action items:

The main problem is how to reduce the time cost or improve calculation speed for real-time rendering. May be achieved by ANN (artificial neural network).

References:

https://unity3d.com/real-time-rendering-3d

09/14/2020 machine learning based Depth Image-Based Rendering (DIBR) algorithm

Xiaoxuan Ren - Sep 28,

Title: machine learning based Depth Image-Based Rendering (DIBR) algorithm

Date: 09/14/2020

Content by: Xiaoxuan Ren

Goals: learn about current algorithm doing 3D rendering

SUMARRY of a research paper (see in reference)

Content:

Two stages of 2D- 3D conversion:

- the estimation of a depth map from color images
- DIBR of a new image to form a stereo-par (multi-view) set of images.

Proposed Algorithm:

The proposed automatic 3D structure estimation algorithm can be formulated as follows. Given a query color image Q, and a database DB composed by cc their corresponding depth maps (RGBD), the **goal** is to estimate the depth map Dest of Q.

The algorithm is divided into two main modules:

- offline pre-processing of DB to speed up the posterior image searches

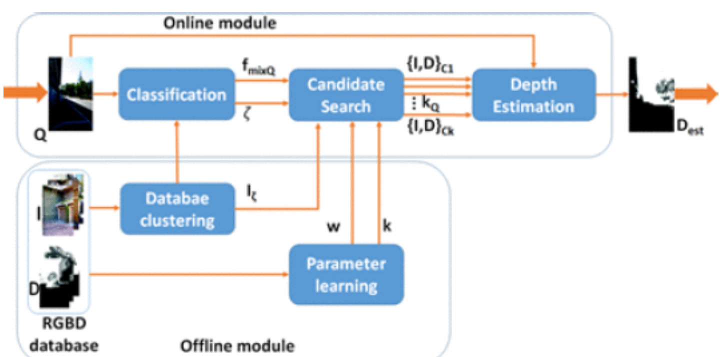
two purposes:

1. to improve the efficiency of the image search strategy by clustering DB according to the structural similarity of the involved color images.
2. to learn the best parameters (weight combination and number of candidate images) for the posterior depth extraction stage.

- online processing to estimate the depth of each incoming Q.

three stages:

1. classification of Q into one (or several) of the classes determined by the DB clusters (i.e. every cluster represents a class).
2. the search of the most similar color images to Q among the members of the resulting class (or classes) is performed.
3. a combination of the depth maps corresponding to the previous selected color images is carried out to obtain Dest.



Computational Times:

COMPUTATIONAL TIMES

Algorithm	Make3D	NYU	Make-NYU	Stereo RGBD 1
Depth Transfer [17]	**92.7**	98.7	101.4	108,3
Depth Fusion [19]	0.83	3.22	4.11	18.16
Adaptive LBP [20]	0.98	3.81	4.93	19.84
Proposed approach	0.17	**0.48**	**0.52**	**1.7**
Offline module	6.9	**7.2**	**7.3**	**8.3**

Evaluation of the computational time in seconds for different state-of-the-art algorithms and for different databases.

Conclusions/action items:

The offline module is run beforehand and involves the main computational cost, alleviating the burden of the online module and making feasible the implem proposed approach in consumer electronics devices such as smartphones or TVs.

The approach achieves similar or higher results to the best algorithms in the state of the art, outperforming them for the most challenging cases, such as th scenes, and for the combinations of indoor and outdoor scenes.

Xiaoxuan Ren - Sep 28, 2020, 4:06 PM CDT

A Novel 2D to 3D Video Conversion System Based on a Machine Learning Approach

José L. Herrera, Carlos R. del-Blanco and Narciso García

A_novel_2D_to_3D_video_conversion_system_based_on_a_machine_learning_approach.pdf(460.3 KB) - download

09/27/2020 3D glass protocol

Xiaoxuan Ren - Sep 29, 2020, 8:37 PM CDT

Title: 3D Glass protocol

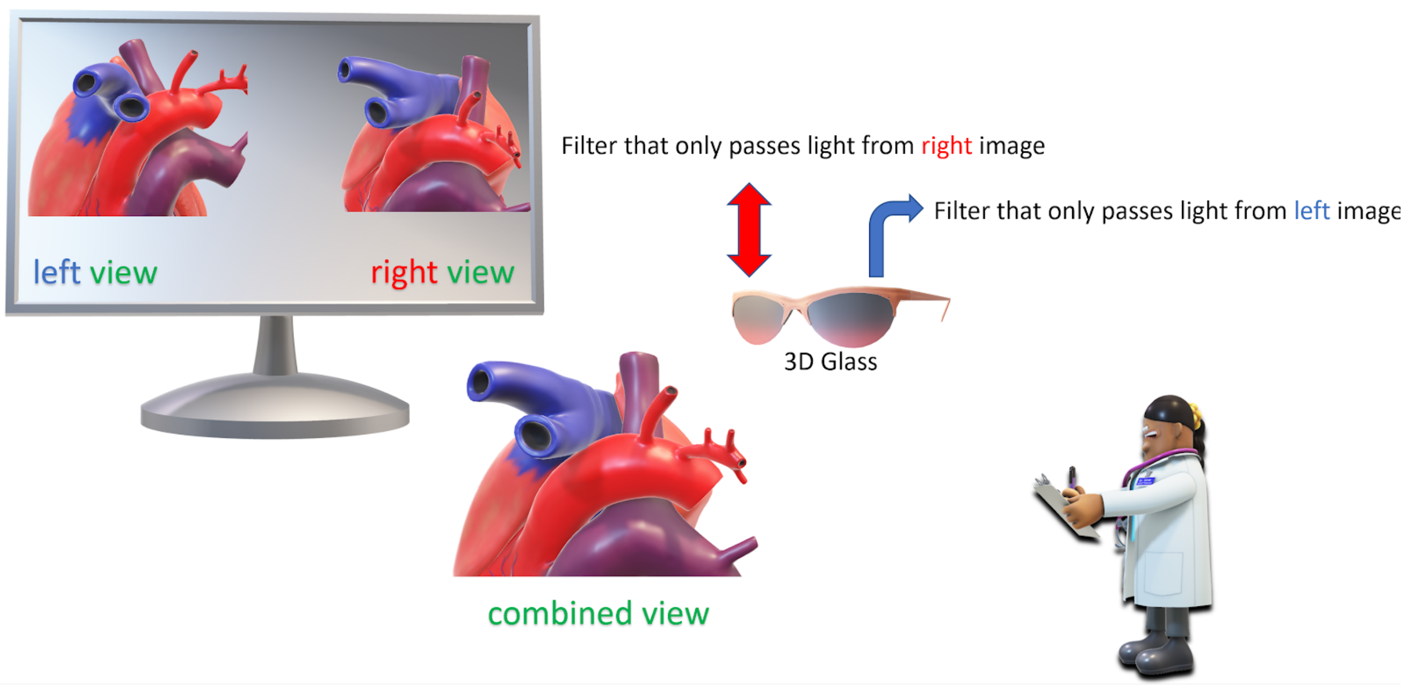
Date: 09/27/2020

Content by: Xiaoxuan Ren

Present: team

Goals: design 1

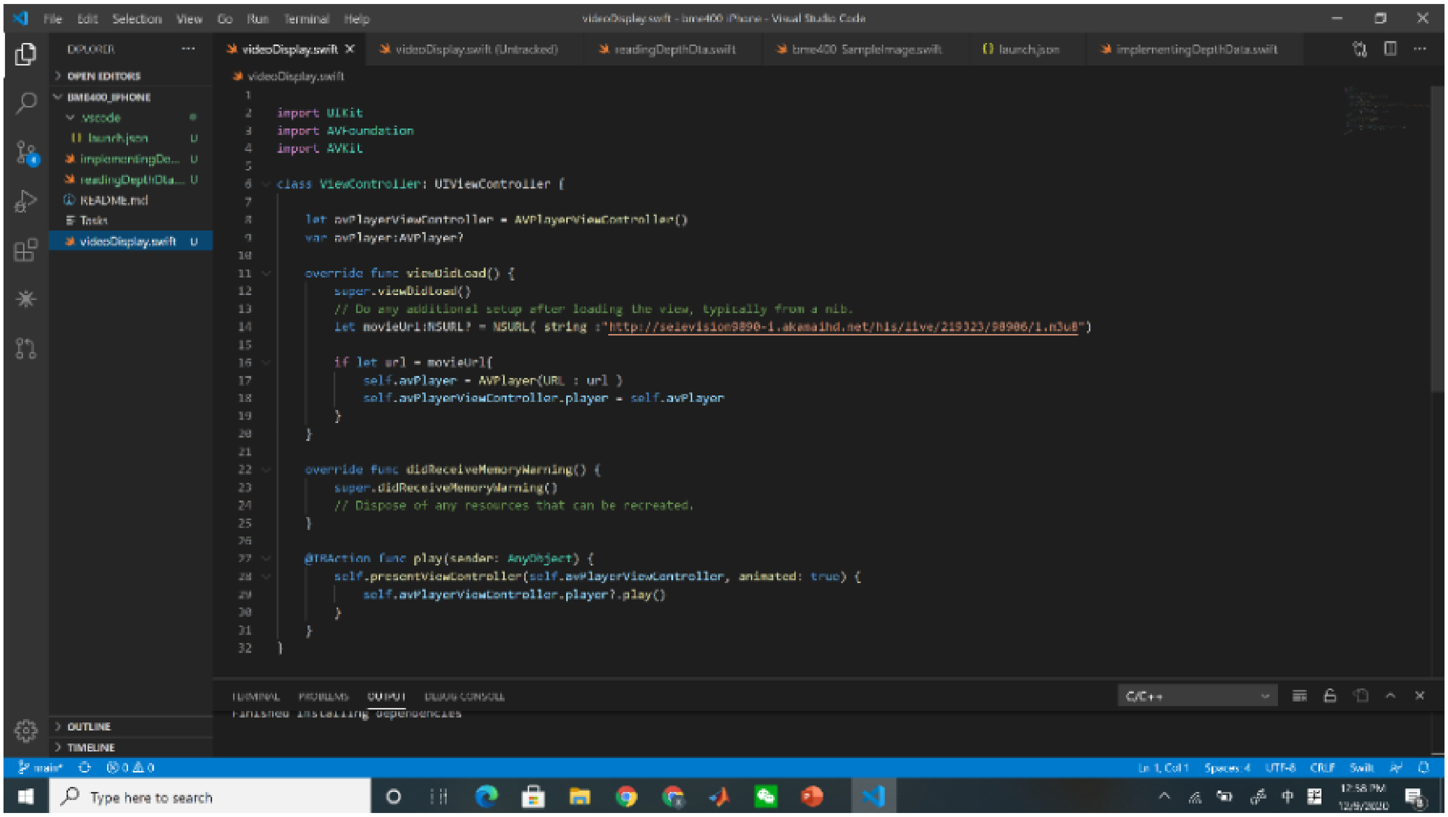
Content:



Conclusions/action items:

Xcode --display video

Xiaoxuan Ren - Dec 09, 2020, 12:59 PM CST



Xcode -- depth extraction

Xiaoxuan Ren - Dec 09, 2020, 1:03 PM

videoDisplay.swift | videoDisplay.swift (Untracked) | readingDepthDta.swift | bme400_SampleImage.sw

readingDepthDta.swift

```swift
static func depthDataMap(forItemAt url: URL) -> CVPixelBuffer? {
  // 1 create a CGImageSource that represents the input file.
  guard let source = CGImageSourceCreateWithURL(url as CFURL, nil) else {
    return nil
  }

  // 2 copy the disparity data from its auxiliary data from the image source at index 0
  let cfAuxDataInfo = CGImageSourceCopyAuxiliaryDataInfoAtIndex(
    source,
    0,
    kCGImageAuxiliaryDataTypeDisparity
  )
  guard let auxDataInfo = cfAuxDataInfo as? [AnyHashable : Any] else {
    return nil
  }

  // 3 extract the orientation
  let cfProperties = CGImageSourceCopyPropertiesAtIndex(source, 0, nil)
  guard
    let properties = cfProperties as? [CFString: Any],
    let orientationValue = properties[kCGImagePropertyOrientation] as? UInt32,
    let orientation = CGImagePropertyOrientation(rawValue: orientationValue)
    else {
      return nil
  }

  // 4  create an AVDepthData from the auxiliary data read in.
  guard var depthData = try? AVDepthData(
    fromDictionaryRepresentation: auxDataInfo
  ) else {
    return nil
  }

```

```swift
  // 5  ensure the depth data is correct format: 32-bit floating point disparity information
  if depthData.depthDataType != kCVPixelFormatType_DisparityFloat32 {
    depthData = depthData.converting(
      toDepthDataType: kCVPixelFormatType_DisparityFloat32
    )
  }

  // 7  apply the correct orientation and return this depth data map.
  return depthData.applyingExifOrientation(orientation).depthDataMap
}
```

Xiaoxuan Ren - Dec 09, 2020, 1:14 PM CST

The most important class, in the iOS SDK, for depth data is `AVDepthData`.

Different image formats store depth data slightly differently.

JPGs store it as a second image within the same JPG. --- use AVDepthData to extract

1. create a **`CGImageSource`** that represents the input file.
2. copy the disparity data from its auxiliary data. from the image source at index 0,
3. extract the orientation to correctly align the depth data,
4. create an **`AVDepthData`** from the auxiliary data read in.
5. ensure the depth data is the format 32-bit floating point disparity information — convert it if it isn't.
6. apply the correct orientation and return this depth data map.

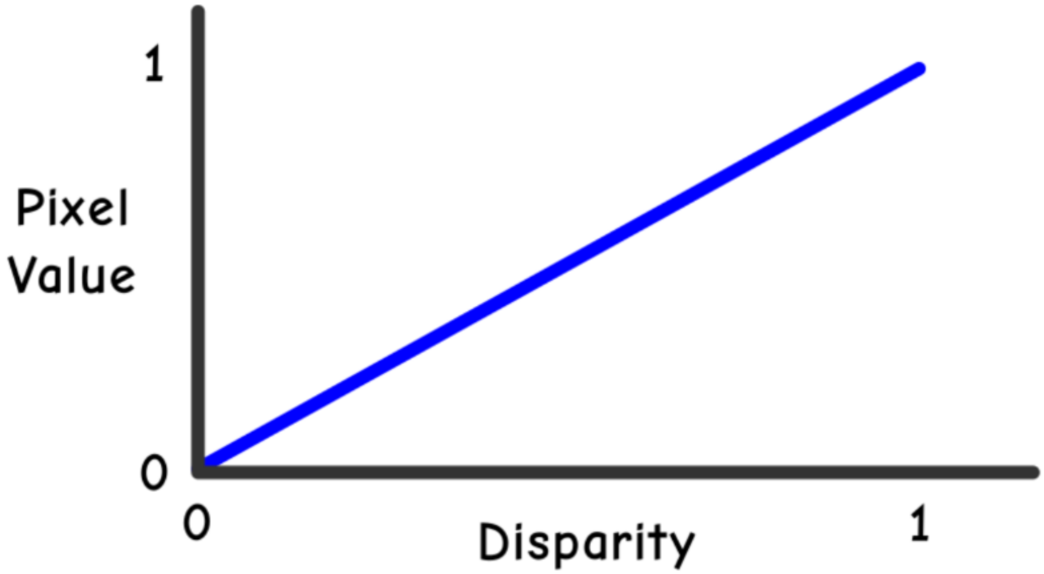
BELOW is the visual representation of the depth data.

The whiter the pixel, the closer it is, the darker the pixel, the farther away it is.

The normalization ensured that the furthest pixel is solid black and the nearest pixel is solid white.

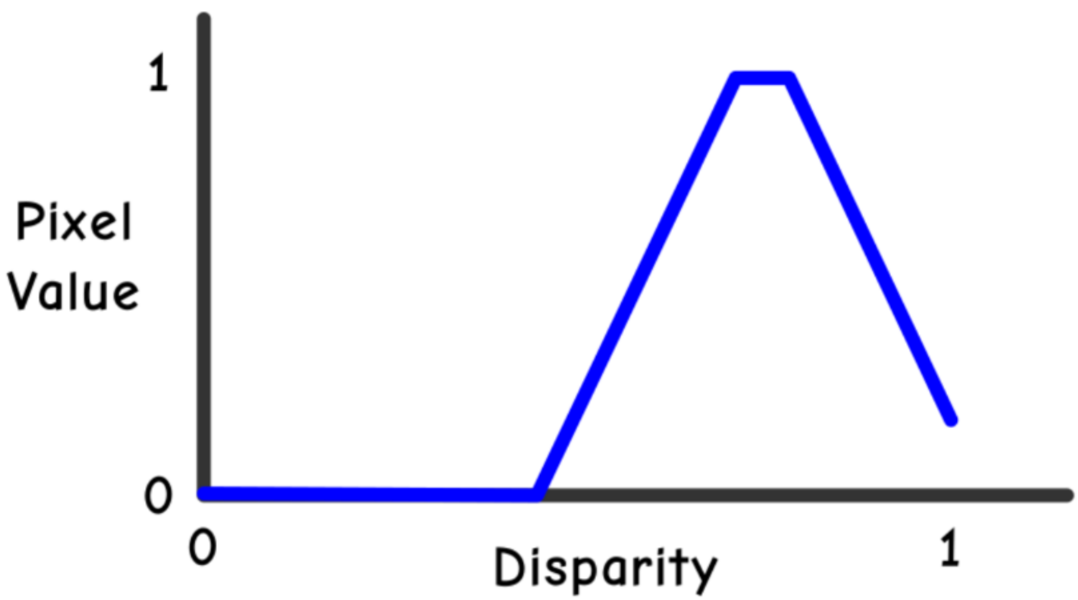
12/9/2020 Read Depth Data

Xiaoxuan Ren - Dec 09, 2020, 1:22 PM CST



The pixel value of depth map image is equal to the normalized disparity.

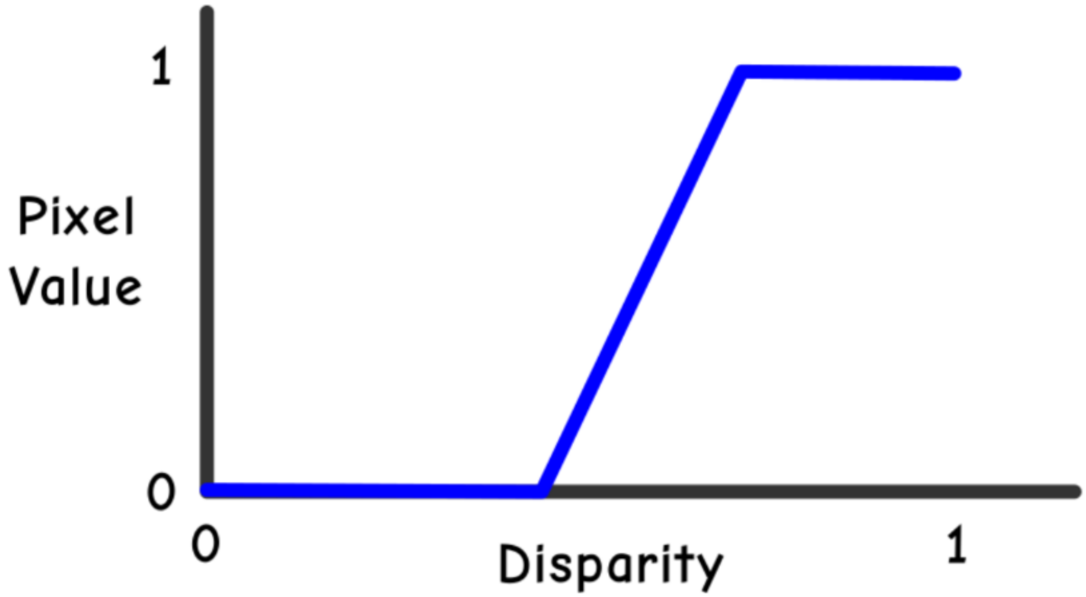
A pixel value of 1.0 is white and a disparity value of 1.0 is closest to the camera. On the other side of the scale, a pixel value of 0.0 is black and a disparity value of 0.0 is farthest from the camera.



This is showing a focal point of 0.75 disparity, with a peak of width 0.1 and slope 4.0 on either side. `createMask(for:withFocus:)` will use some funky math to create this function.

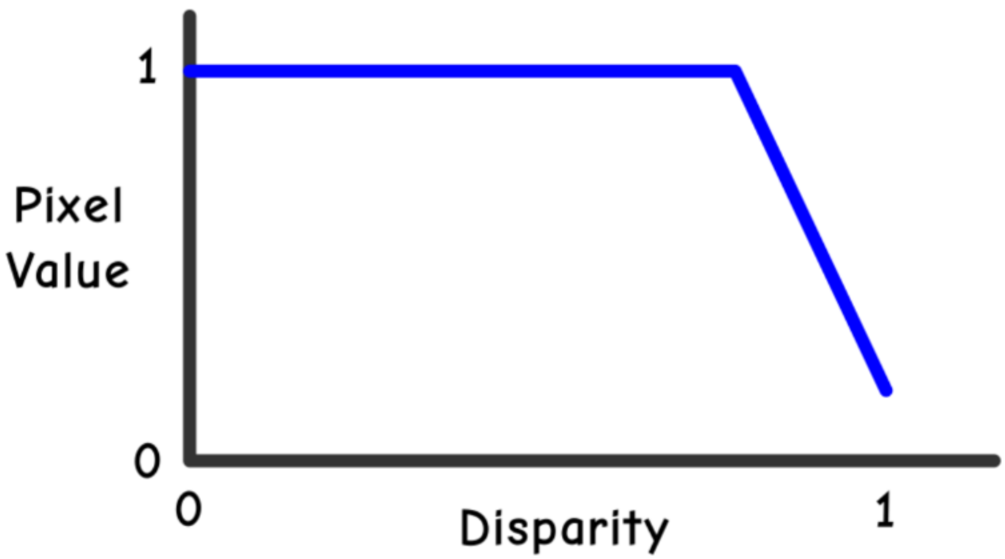
This means that the whitest pixels (value 1.0) will be those with a disparity of 0.75 ± 0.05 (focal point ± width / 2). The pixels will then quickly fade to black for disparity values above and below this range. The larger the slope, the faster they'll fade to black.

set the mask up in two parts — combine the left side and the right side.



This filter multiplies all the pixels by the slope s1.

The larger s1 is, the steeper the slope of the line will be. The constant b1 moves the line left or right.



combine the masks by using the CIDarkenBlendMode filter, which chooses the lower of the two values of the input masks.

2014/11/03-Entry guidelines

John Puccinelli - Sep 05, 2016, 1:18 PM CDT

Use this as a guide for every entry

- Every text entry of your notebook should have the **bold titles** below.
- Every page/entry should be **named starting with the date** of the entry's first creation/activity, subsequent material from future dates can be added later.

You can create a copy of the blank template by first opening the desired folder, clicking on "New", selecting "Copy Existing Page...", and then select "2014/11/03-Template")

Title: Descriptive title (i.e. Client Meeting)

Date: 9/5/2016

Content by: The one person who wrote the content

Present: Names of those present if more than just you (not necessary for individual work)

Goals: Establish clear goals for all text entries (meetings, individual work, etc.).

Content:

Contains clear and organized notes (also includes any references used)

Conclusions/action items:

Recap only the most significant findings and/or action items resulting from the entry.

2014/11/03-Template

John Puccinelli - Nov 03, 2014, 3:20 PM CST

Title:

Date:

Content by:

Present:

Goals:

Content:

Conclusions/action items: