



Real Time Green Screen & Blurring



Gustavo Fonseca
Jason Vasko





Introduction

Goal

Real Time Green Screen and Portrait Mode using the tablets camera

Algorithm

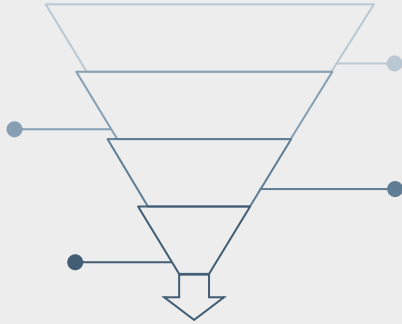
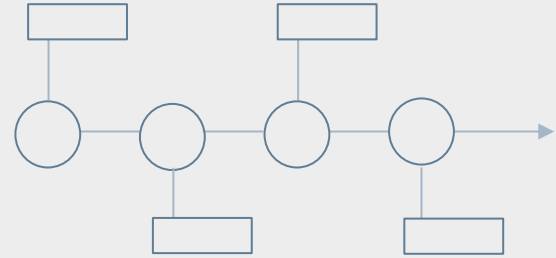
Uses Foreground & Background segmentation to identify changing pixels

App

Allows users to pick between the two modes and allows for different background for green screen mode.



Algorithm





Algorithm: Overview

Main Goal

- Identify if a pixel is in the foreground or background
 - Foreground refers to things that are moving

4 Key Steps

- Cluster Matching
- Adaptation
- Normalization
- Classification



Cluster Matching

- Each pixel has N number of clusters assigned to it
- matches the new pixel values to the nearest cluster
- Compare the shortest distance to a predefined threshold T (i.e. 10)



Cluster	Value	Weight
0	55	0.5
1	30	0.3
2	75	0.05
3	20	0.15

Adaptation: No Match

Update the clusters and weights

- Smallest Weight Cluster Replaced
- New Weight Set to the small value (i.e. 0.01)

New Pixel (220)



Cluster	Value	Weight
0	55	0.5
1	30	0.3
2	75 220	0.05 0.01
3	20	0.15

Adaptation: Match Found

Update the clusters and weights ($L = 10$)

- Update all cluster weights based on the equation:
- Centroid Value of the matched cluster:

$$w'_k = \begin{cases} w_k + \frac{1}{L}(1 - w_k), & \text{if } k = M_k, \\ w_k + \frac{1}{L}(0 - w_k), & \text{if } k \neq M_k, \end{cases}$$

$$c'_k = c_k + \frac{1}{L}(x_t - c_k)$$

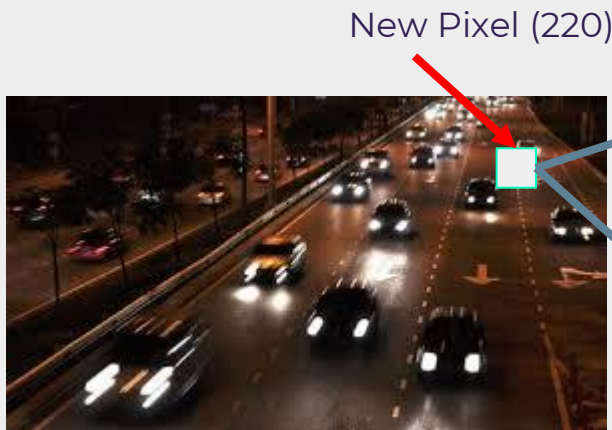
New Pixel (15)



Cluster	Value	Weight
0	55	0.5 0.45
1	30	0.3 0.27
2	75	0.05 0.045
3	20 19	0.15 0.235

Normalization

- normalize weights to sum to one
- Weight Represents the probability of that value being part of the background



Cluster	Value	Weight
0	55	0.5 0.52
1	30	0.3 0.31
2	75 220	0.05 0.01
3	20	0.15 0.16

Classification

- sum all weights that have a weight higher than the matched cluster's
- Can make a threshold to determine foreground or background (i.e. 0.5)



New Pixel (220)

$0.52 + 0.31 + 0.16 = 0.99$
99% chance of foreground

Cluster	Value	Weight
0	55	0.5 0.52
1	30	0.3 0.31
2	75 220	0.05 0.01
3	20	0.15 0.16

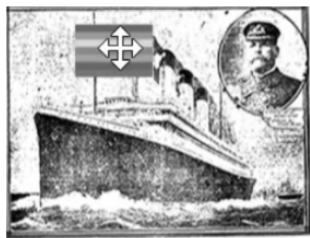
Post-Processing

- Lot of Noise in initial segmentation
- Used a Open Filter for this: Erode and then Dilate Filters

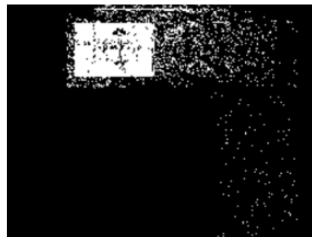
$$x = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \text{dilate}(x, S) = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{erode}(x, S) = 0 \quad (\text{all zeroes})$$

Results of Project Lab



(a) Gray-scale Video Frame



(b) Output before post-processing



(c) Output after post-processing

Output

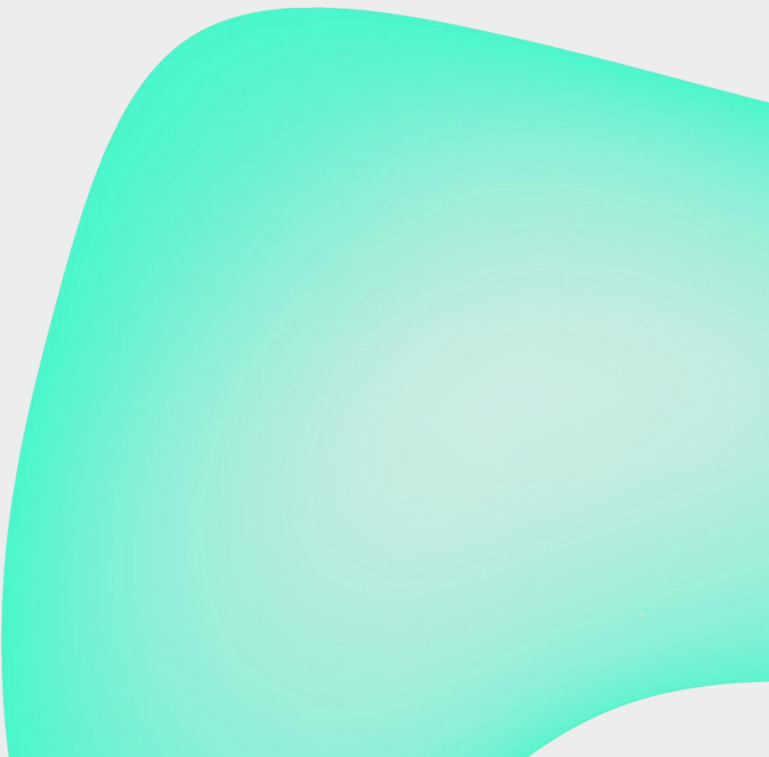
- Identified moving objects
- Extremely noisy
- Very susceptible to exposure changes
- Extremely Slow (~10 Frames/Sec)

Things to improve

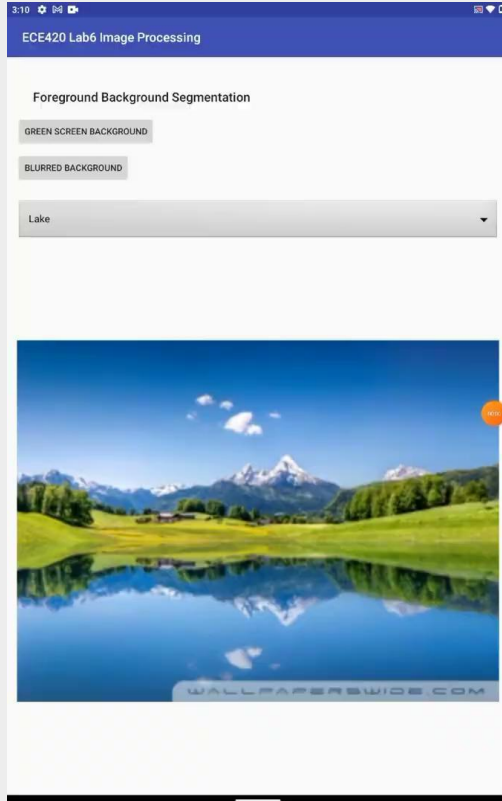
- Optimize runtime of algorithm
- Lock video exposure
- Fine tune the post-processing



App Features: Green Screen

- User Can pick between 3 different backgrounds: Mountains, Forest, and Beach
 - All pixels found to be foreground will be unchanged, otherwise the pixel is replaced
- 

App Example: Green Screen



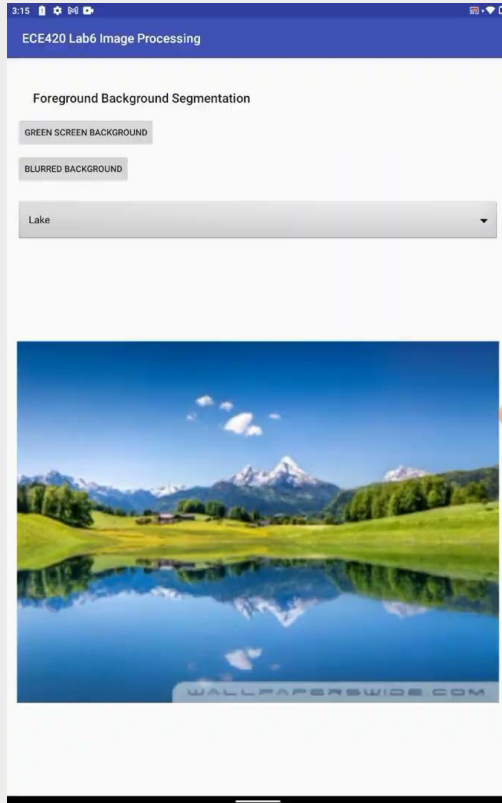
App Features: Portrait Mode

- All pixels found to be foreground will be unchanged
- If the pixel is from the background, then it will be replaced with a blurred version
 - Used a 5x5 Box Filter:

$$\frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

App Example: Portrait Mode



App Results

- The app is able to work in real time
 - Using grayscale and small resolution for faster results
- Built every function from scratch
 - Including all post processing
- User can pick between the two modes in the home screen

Conclusion

- App accomplished every goal of MVP
 - Has both Portrait and Green Screen Modes
 - Applies the algorithm to a video taken from the tablet's camera
 - Allows the user to choose either method at start up
- The Stretch goals we completed were
 - Real Time Application of the algorithm to live video feed
 - Multiple background choices
 - Implemented every function without the use of openCV



Thank You!

