CSE 408/598 Project, Phase #1, group 6

Sreenivas Shenoy

Gabriel Nunes Crispino

Jacob Pruitt

James Perry

Derek Norman

Abstract

In this project work was done to learn more about video processing, color models, and some of the tools and libraries available. Python and OpenCV were used to create two programs. The first gets information from the user to create a color map in a color space, scales the colors from -1 to 1, then outputs the color map to a file. The second program gets a video and allows the user to select 2 frames. These frames are converted to grayscale, then a comparison of the difference between the images is calculated. A color map is added to the image after a scaling is done, to help see the differences between the frames.

Keywords

Color map, color model, RGB, XYZ, Lab, Luv, YCrCb, HLS, HSV, grayscale, scaling color values.

Introduction

OpenCV is a library for different programming languages that provides tools to manipulate images, videos, and other forms of media.

Color models are different standers to represent colors or differences between colors.

A color map can be used to represent a set of colors.

A color instance is representation of a color.

Delta, or ∆, is the difference or change from one object to another.

The goals were to be able to generate a color map based off of user inputs, and to compare 2 frames from a video, scale the differences in gray values, and then apply a color map to the image.

An assumption that was made was the color models that could be used for part 1. Users can select from seven different color models and the program accepts and scales the inputs based on the selected model. Another assumption was in the intervals used when making the color map. Some colors may not be fully represented due to the how colors were assigned. [-1, -0.5), [-0.5, 0), [0, + 0.5), [+0.5, +1].

The solution for part one was to generate a color map based on a given color model by scaling each of the 3 color instances from -1 to 1. The choice of the color model affects how each color instance is scaled.

Part two was implemented by using OpenCV ‘VideoCapture’ object to manipulate the video. [1] The method ‘read()’ was used to access frames of the video. The method cvtColor was used to convert the frames to grayscale.[1] The frames were compared with each other and saved as a new image called ‘gray\_delta’. A file containing color map information is then applied to the gray\_delta image. All of the images are displayed for the user to view.

Interface Specification:

Part1

1. RGB 2. XYZ 3. Lab 4. Luv 5.YCrCb 6. HLS 7. HSV

|  |  |
| --- | --- |
| Prompt: | input |
| Enter value of C0, the format should be (0-255,0-255,0-255): | 0,0,0 |
| Enter value of C1, the format should be (0-255,0-255,0-255): | 0,0,255 |
| Enter value of C2, the format should be (0-255,0-255,0-255): | 255,255,255 |
| Enter number of bits (from 2 to 8): | 8 |

A colormap with 256 colors will be created

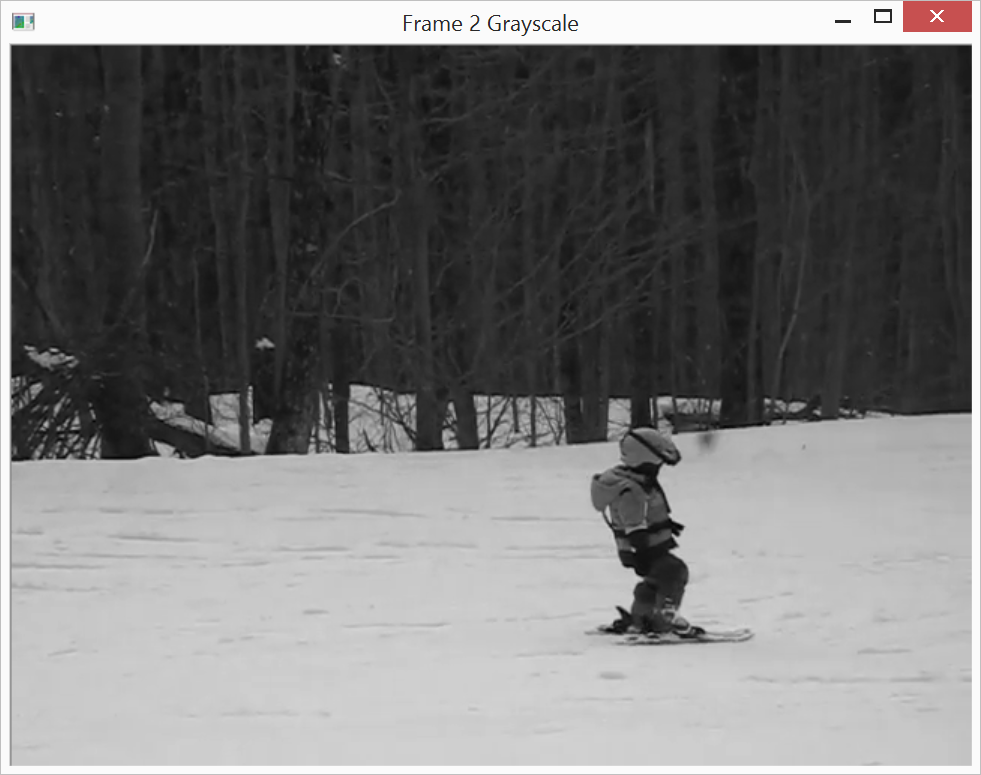
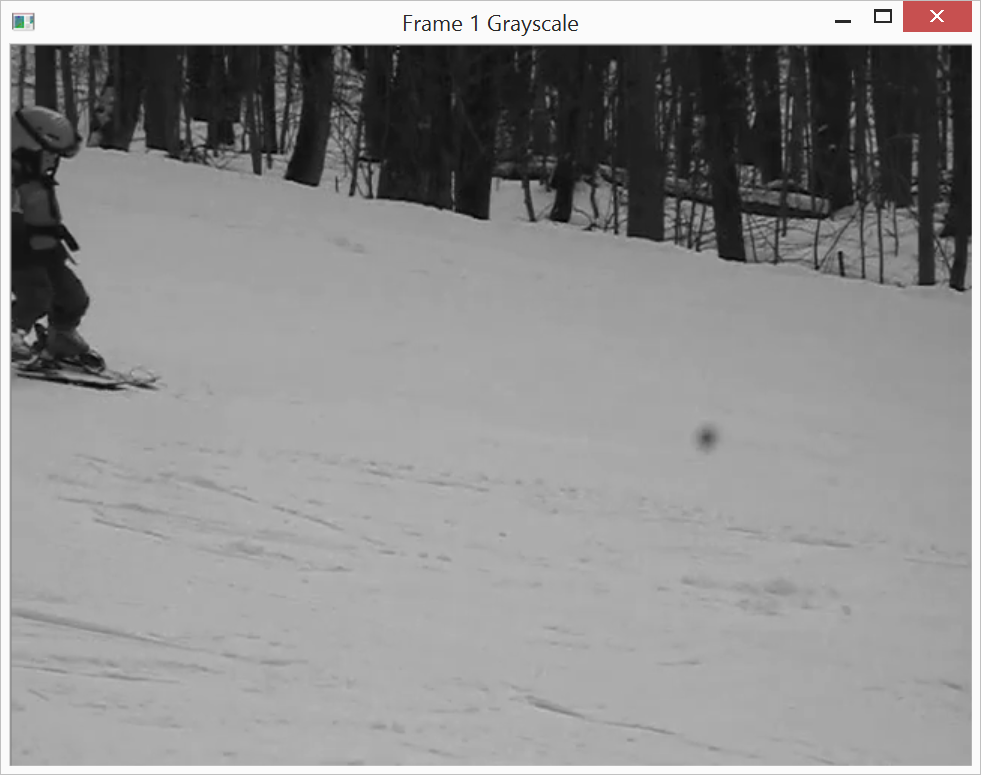


A txt file is also created with the color map information. This information can be used in part 2.

Part2:

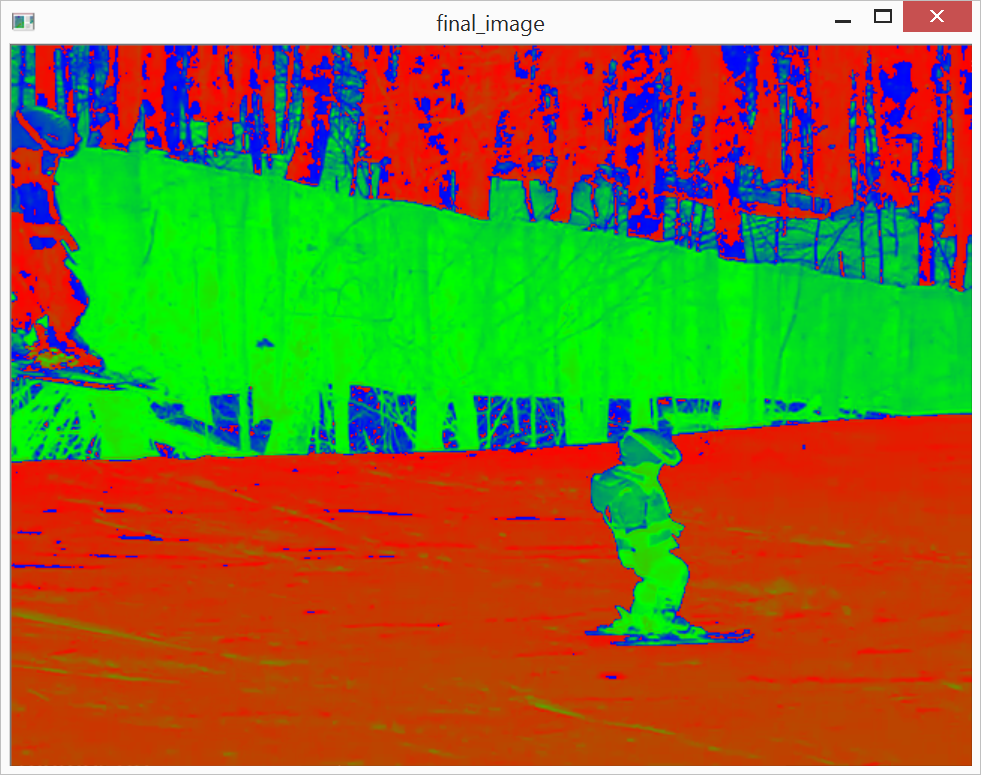
Absolute paths must be used when giving the path to the videos. Also, the ‘\’ should not be used as the last character of the folder path.

|  |  |
| --- | --- |
| Prompt: | input |
| Please enter the path where video files are located: | C:\Users\azfut\_000\PycharmProjects\408\samples |
| Directory set to: C:\Users\azfut\_000\PycharmProjects\408\samples is this okay? Y/N | y |
| Enter the name of the file you wish to read: | 5.mp4 |
| C:\Users\azfut\_000\PycharmProjects\408\samples/5.mp4 has been selected for processing is this okay? Y/N | Y |
| Number of frames of the video: 880 |  |
| Enter the first frame to compare: | 5 |
| Enter the second frame to compare: | 700 |





|  |  |
| --- | --- |
| Please select a colormap for processing: | colormaptextfile.txt |



System Requirements

Suggested is a Linux (debian) environment. Python 2.7 with numpy installed, and OpenCV 2+ are required.

Related work

Color models are ways of representing colors. Each model has a different format and values in other models can be represented on an XYZ coordinate grid. XYZ represents all the humans can see. Color models are designed with the human eye in mind. The limits of colors and intensities the humans can perceive are taken in to account. Color that human can see or that are not perceived as well, are not usually in color models, or have less effect in the model. Humans have 3 kinds of cones in their eyes that are sensitive to either red, green or blue, these colors map to XY and Z [2]. RGB is a model that is good at capturing what humans can see, and is a liner transformation from XYZ [2]. RGB can be used to display images and videos, but other models can be used depending on the needs of the user. The number of bits that need to be used or should be used, how much space can be given for media, and resources available to encode and decode are important questions when determining a color model.

Conclusion

OpenCV and python were used to learn about color models and to create a color map from user inputs any of 7 different color models. Color values have been scaled to a -1 to 1 values. Given videos were analyzed and 2 frames were extracted, converted to grayscale, and compared. A color map was then applied to the delta image to show the differences between the gray values as colors from the map.

Bibliography

[1] "OpenCV: Changing Colorspaces." OpenCV: Changing Colorspaces. OpenCV, 18 Sept. 2015. Web. 20 Sept. 2015.

[2] Li, Ze-Nian, Mark S. Drew, and Jiangchuan Liu. *Fundamentals of Multimedia*. Cham:Springer, 2014.Print.

Appendix

Team Member Contributions

Sreenivas Shenoy

Part 1 code

Part 1 testing

Gabriel Nunes Crispino

Part 2 code

Part 2 testing

Jacob Pruitt

Part 1 code

Part 1 testing

James Perry

Part 2 code

Part 2 testing

Derek Norman

Report composition

Code review