

Phase 1- Report

Mukund Manikarnike, Aaron Leyesa, Pawan Mahalle, Christopher Mendez and Matthew Mendez

Abstract:

In this project our group experimented with video processing libraries and color models by implementing them into a python script. We used 7 different color spaces. Using those color spaces we made a color map to three color instances. We then extracted two specified frames from a video file and displayed both in grayscale. The two grayscale images were used to make a difference image and the difference image was visualized using the selected color map.

Keywords:

Color space, Color model, Color instance, Color map, Color opponents, Grayscale, RGB, XYZ, Lab, YUV, YCbCr, YIQ and HSL.

Introduction:

In this project we created color maps which are used to transform one of the colors of a source image to another color. The color maps range in value from -1.0 to 1.0 and map each number within the range to a color instance. A color instance is a specific combination of the three or four colors used in a color model. The color model mathematically describes colors in tuples or quadruples of numbers. The numbers represent the intensity of the colors.

We used seven different color spaces. A color space is an organization of colors used in both analog and digital representations which allows for reproducible colors. The color spaces we used were RGB, XYZ, Lab, YUV, YCbCr, YIQ, and HSL. RGB uses the colors Red, Green and Blue. XYZ has imaginary colors which the average human cannot see. The real colors that exist in XYZ are the ones used in the subset of RGB.

The Lab color space includes all perceivable colors and uses color-opponents. Color opponents are broken up into three different channels. Red versus green, blue versus yellow, and black versus white. The black versus white channel is used for the lightness of the color-space, or the L dimension. The two other channels, red versus green and blue versus yellow, are dimensions a and b.

In the YUV color space there are two chrominance components (UV) and one component for brightness or luma. The YCbCr color model is used for digital video and are derived from YUV. The Cb and Cr components are deviations from gray on the blue-yellow and red-cyan axes.

The YIQ color space, which is used by the color TV system in North America, is similar to the YUV color space but uses different color opponents. The I in the YIQ color space is the changes between orange and blue while the Q is the change between purple and green.

The HSL color space is a bit different from the other color spaces since it uses a cylindrical-coordinate representation instead of using a cartesian approach. HSL stands for hue, saturation and lightness. The degree in the circle of the cylinder represents the hue the distance from the center of the circle is the saturation and the height of the cylinder is the range of brightness from black to white.

We assumed the input for the color instances would be in a 3 channel format. For problem 4 we assumed that a color map would be given as input to the script and is to be placed in the same folder as the video file that we are extracting from.

Description of the proposed solution/implementation:

For the color map portion of the project, our solution was pretty straightforward. A user is prompted for three different colors to use for the color map in RGB format. Once this is done, a color map is created by converting the RGB values to the various types of color spaces. The size of the color space is specified by the user and this size is then implemented by creating a color map of n bits, where the number of colors is 2^n .

For the differentiation part of the project, the difference between the two image frames in a video is output to a file, and this difference image can be mapped to a color map specified within a text file. This difference image is created using the OpenCV library, which has a function for this already defined. The result of this image is a grayscale image, and this grayscale image can be mapped to the color map through the use of our color map algorithm which basically is a mapping function from the grayscale intensity (white to black \rightarrow 0 to 255) scaled down to -1 to 1 within the color map.

Interface specifications (with focus on the goal and problem specification):

For problem 3, the user is prompted to make a choice regarding the color space between the 7 different types of color spaces. Once they enter this, they are prompted for the R, G, and B values of 3 colors to use as the basis for the color mapping. Then, they enter the number of bits for the color map. The number of bits corresponds exponentially to how many different colors will be available within the color space. For example, a bit number specification of 4 will result in a color space 2^4 colors. These colors are then spread across the color map and output to a text file where they can be parsed in the problem 4 part of the project.

For problem 4, the user is prompted for the video path and file name. They are then given the number of frames in the video and can choose between any frame 1 \rightarrow frameNumber for their frame numbers. These frames are then differentiated and the user is asked if they would like to map the difference with a color map file. If the user says no, the program ends, and only the difference image is output in the same directory as the input video file.

If the user says yes to mapping the difference between the two frame images, then the user is prompted for the name of the color map file. The color map file is then parsed and used as a color map, which first maps the images to grayscale and then maps them to specified color map. The color mapped differentiation is output to the folder of the color map file.

System requirements/installation and execution instructions:

We used Python 2.7 with the OpenCV 3.0.0 libraries. Other libraries we used were:

- Numpy
- Matplotlib
- DateUtil
- Pyparsing

Conclusions:

There were 2 steps to this phase of the project.

1. Creating the color map. This step taught us how a 3 dimensional space of colors can be viewed as a set of lines in 3 dimensional space all of which can be partitioned into as many instances as desired, the number of instances desired being directly proportional to the number of colors one can represent. We also learned here that by creating and visualizing this color map, we are actually representing a certain color in a chosen color space such as XYZ, Lab etc in RGB. The outputs clearly illustrate this as one can notice how shades of red, green and blue.
2. Extracting the difference image of 2 frames in a video and applying the color map to the image. This step helped us understand the issues in reading frames from videos. The OpenCV library would read different images when reading the video at different times Hence, the difference would be different at different reads. Also, we saw that when the order of the difference in frames is reversed, the difference image when mapped using the color map has regions which have colors reversed. This is because the pixel difference in grayscale only changes in sign and not in magnitude and when such a pixel is mapped to the color map, it gives us such a difference.

Bibliography:

- Cruse, Phil. Introduction to Colour Spaces. N.p., 12 June 2015. Web. 20 Sept. 2015. <www.colourphil.co.uk/lab_lch_colour_space.shtml>.
- "EasyRGB - The Inimitable RGB and COLOR Search Engine!" EasyRGB - The Inimitable RGB and COLOR Search Engine! N.p., n.d. Web. 21 Sept. 2015. <<http://www.easyrgb.com/index.php?X=CALC>>.
- "EasyRGB - The Inimitable RGB and COLOR Search Engine!" EasyRGB - The Inimitable RGB and COLOR Search Engine! N.p., n.d. Web. 21 Sept. 2015. <<http://www.easyrgb.com/index.php?X=MATH>>.
- "HSL to RGB Color Conversion." HSL to RGB Conversion. N.p., n.d. Web. 21 Sept. 2015. <<http://www.rapidtables.com/convert/color/hsl-to-rgb.htm>>.
- Online Color Converter. Picture to People, n.d. Web. 21 Sept. 2015. <http://www.picturetopeople.org/color_converter.html>.
- Schweyer, Manfred. "Color Conversion." Color Conversion. Equasys GmbH, n.d. Web. 21 Sept. 2015. <<http://www.equasys.de/colorconversion.html>>.
- Vishnevsky, Eugene. "Color Conversion Algorithms." Color Conversion Algorithms. Ed. Nan C. Schaller. Rochester Institute of Technology, n.d. Web. 21 Sept. 2015. <http://www.cs.rit.edu/~ncs/color/t_convert.html>.
- Wright, Chris. "YUV Colorspace." YUV Colorspace. N.p., 2002. Web. 21 Sept. 2015. <<http://softpixel.com/~cwright/programming/colorspace/yuv/>>.

Appendix:

Aaron Leyesa	Worked on problem 3 as detailed in the rubric, the color space/map implementation/interface
Pawan Mahalle	Worked on problem 3 as well with Aaron
Mukund Manikarnike	Implemented the interface and logic for problem 4 as detailed in the rubric, the video frame extractor and grayscale/difference visualizer
Christopher Mendez	Wrote the report and analysed problem 4 for any bugs/issues during group meetings
Matthew Mendez	Wrote the report and analysed problem 4 for any bugs/issues during group meetings