

ASSIGNMENT 3

NAME:SAI
SREEBHAGI

CLID: sxb8547

Question1:

The Shape of the L*a*b* Color Space

Unlike the RGB or HSV spaces, the L*a*b* space does not have a regular shape . In this assignment, explore the shape of the L*a*b* color space.

Divide the L*a*b* space into 500 bins, as follows:

5 bins in the L* dimension

10 bins the a* dimension

10 bins the b* dimension

Map the colors from a standard RGB cube to the bins Find out how many bins in the L*a*b* space are occupied and how many are always empty

Optionally, you can find a

creative way to display or illustrate the L*a*b* (as a 3d volume or as slices). Write a complete report of your work, including your documented source code and results. Submit your report and your source code as attachments.

Solution:

Source code:

```
#include <opencv2/core/core.hpp>

#include <opencv2/imgproc/imgproc_c.h>

#include <opencv2/highgui/highgui.hpp>

#include <iostream>

#include <string>

using namespace std;

int main()

{

// loading an image

IplImage* image = cvLoadImage("images/test.jpg", 1);

//condition if the image is not found
```

```

if (!image)
{
    cout << "file not found" << endl;
    exit(0);
}

// Creating the image1 equivalent to the size of original image
IplImage* image1 = cvCreateImage(cvSize(image->width, image->height),IPL_DEPTH_8U, 3);

//converting the color image to l*a*b* color space
cvCvtColor(image, image1, CV_BGR2Lab);

// creating a 3D histogram for an image
int histogram[5][10][10];

// Initializing the count for no. of pixels occupied in each bin
int count = 0;

// processing the entire image by refering each row & column in a image
for (int row = 0; row < image1->height; row++)
{
    for (int column = 0; column < image1->width; column++) {
        int l, a, b;

        // initializing the channels
        int step,channels=3;

        //defining the pointer for stepping into each row
        step=image1->widthStep;

        // indexing the bins in each channel
        l = image1->imageData[step * row + channels * column + 0] / 51.0;
        a = image1->imageData[step * row + channels * column + 1] / 25.5;
        b = image1->imageData[step * row + channels * column + 2] / 25.5;
    }
}

```

```

// checking whether any bin in the histogram is occupied with l*,a* and b* color spaces or not
if (histogram[l][a][b] != 1)
{
// defining the occupied bins in histogram to '1'
    histogram[l][a][b] = 1;
// incrementing the count
    count += 1;
}
}
}

// calculating the unoccupied bins by just subtracting the occupied bins from total no. of available bins
int unoccupied = 500 - count;

// printing the no. of unoccupied and occupied(count) bins in a histogram
cout << "Number of unoccupied bins: " << unoccupied << endl;
cout << "Number of occupied bins: " << count << endl;
}

```

Description:

In this, I first loaded the color image by using `cvLoadImage()`. I used `cvCvtColor()` to convert BGR color image to $l^*a^*b^*$ color space.

By using 3D array, I formulated a histogram for this $l^*a^*b^*$ color space. In

histogram, I indexed the bin by just dividing each channel with corresponding

channel's bin size. Since, in the problem it is given that we need to formulate 500 bins in such a way that we have 5 bins for l^* channel and 10 bins for a^* , 10 bins for b^* so, to get the individual bin size in each channel, I divided the channel size of 255 by corresponding partitions as follows:

For l^* channel, bin size = $255/5=51.0$

For a^* channel, bin size = $255/10=25.5$

For b^* channel, bin size = $255/10=25.5$

Indexing the bins tells how many pixels are there in each bin. Finally, I printed the count of occupied and unoccupied bins in a histogram.

Tools used:

- 1) OpenCv 2.4.6.0
- 2) Microsoft visual studio 2010

Input:

I took original raw .jpg,.gif,.png images one at a time as an input to my program. I placed all the image files int the following folder: images/test.jpg

Outputs:

- 1)Image1:test1.jpg

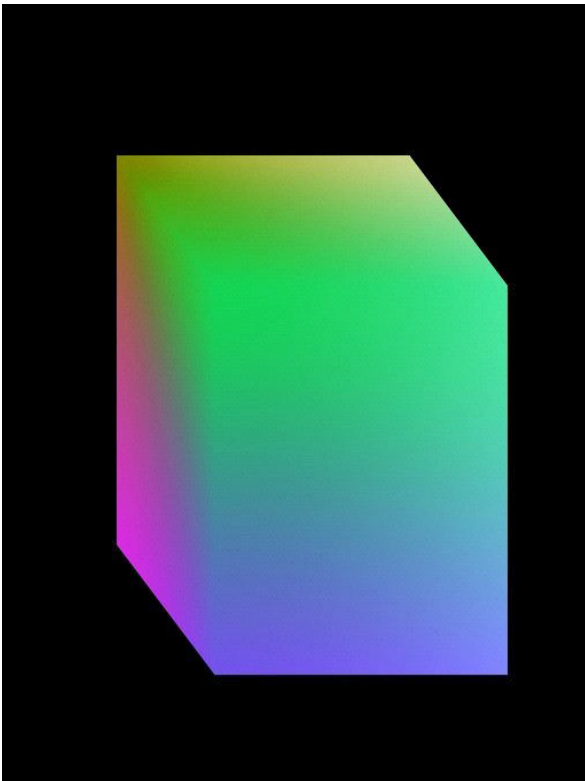


output:

Number of unoccupied bins: 336

Number of occupied bins: 164

2)Image2:lab.jpg



output:

Number of unoccupied bins: 425

Number of occupied bins: 75