**Name: Pravin Reddy Sereddy**

**Email: spravin1@binghmaton.edu**

**Purpose of the Project:**

The main purpose of the project is getting the knowledge of how to compressed and decompression the image data and reduce the memory size of the file without losing the image details. And to get the knowledge of detecting the lines and circles in the given image. So, we can get reliable good images for different types of methods.

I have used 1d DCT transformation, RGB to HSI image conversion, 2d DCT transformation, 2D IDCT transformation, Hough transform methods

And all these methods, implementations and results are explained in the below section

**Programs required environment setup and Running the Programs:**

There are 3 program files in this zipped file

**P1.cpp** is a 1d DCT compression for the given arrays for 1st question in part – B

**P2.cpp** is to convert the image to HSI image and apply 2d DCT transformation for the given image to compress the image and apply the 2d IDCT image transformation.

**P3.cpp** is to detect the region of interest in the image such as the straight lines and circles in the image.

**Required Environment setup:**

* Opencv 3.4
* Linux based system with g++ compiler for compiling the c++ programs

**Running the program:**

Select the above program required and compile the program in the terminal with above environment

**For compiling:**

g++ -ggdb filename.cpp -o filename ‘pkg-config –cflags –libs opencv’

**For running:**

./filename image

For program p1

./a.out

For program p2

./p2 image

For program p3

./p3 build.bmp disk.bmp

**Programming Design**

**1. 1d DCT transform method**

* I have used the 1d DCT formula to convert the 8 element array and 16 element array to transform the data in the array
* formula is

\begin{displaymath}
F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1}
\Lambda(i).cos\left[
\frac{\pi.u}{2.N}(2i+1)
\right]f(i)\end{displaymath}

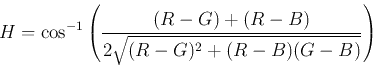
* by applying this formula we get the compressed data

**2. RGB to HSI image conversion**

* take the given image and take each pixel and we can sub divide the pixel into three components and R value , G value, B value
* from these values we use the formula to convert in to hue, saturation and intensity
* for the intensity value I we use this formula

\begin{displaymath}I=(R+G+B)/3 \end{displaymath}

* for the hue value H we use



* for the saturation value S we use

\begin{displaymath}S=1-3\;\min(R, G B)/I \end{displaymath}

* from these HSI values we output the image

**3. 2d DCT transform method**

* Take the image and load the image data in the program
* Take each pixel value and apply the DCT algorithm to those pixel values
* The formula for the 2d DCT compression is

\begin{displaymath}
F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}}
\left(\frac{...
 ...}(2i+1)
\right]cos\left[ \frac{\pi.v}{2.M}(2j+1) \right].f(i,j)\end{displaymath}

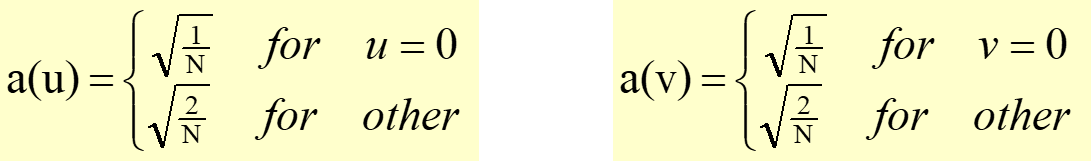
\begin{displaymath}
\Lambda(\xi) = \left\{ \begin{array}
{ll} \frac{1}{\sqrt{2}} & {\rm
for}
\xi = 0 \ 1 & {\rm otherwise}\end{array} \right.\end{displaymath}

* After applying this formula to the data, we get the DCT transformed values.

**4. 2d IDCT transform method**

* In this method we take the DCT values and convert into the gray level pixel values
* Take the dct values and apply this IDCT transformation formula
* The formula for the IDCT is





* After applying this formula to the data, we get the all IDCT transformed values

**5. Detecting the region of intertest method**

* Take the edge image and transfer all the pixel values
* Detect all the straight lines through the parametrized space corresponded to a point in the image space
* Convert the x, y values to the polar coordinate system
* Collect the all the lines and circle data into bins and select the only lines if the bin value is greater than the 80

**6. Detecting the region of intertest method**

* Take the given image and apply the canny edge detection for the image
* After applying the canny edge detection, we get the edge image
* And apply the Hough transform to detect the circles and the straight lines
* And map the all those straight lines and the circles on the original image to see the result

**Results:**

****

1. for the array 10 11 12 11 12 13 12 11 the dct values are

32.52 -1.28 -1.30 0.44 -1.41 -0.30 0.54 0.25

for the array 10 -10 8 -7 8 -8 7 -7 the dct values are

0.35 4.25 0.34 5.04 2.47 8.38 1.76 20.38

1. for the 16-number array transformation is

23.25 21.77 -3.91 -6.90 -0.67 6.03 -3.25 -2.5 0.75 3.15 -6.14 1.60 1.63 5.49 -4.23 13.56

1. The result (b) block size 16 has the greater compression than the 8 size block because if you add the positives and negatives values we get less for the 16 block size DCT than the 8 block size
2. The original image f1 (“basel.bmp”)

A boat sitting on top of a building

Description generated with high confidence

1. The intensity imageA black and white photo of a city

   Description generated with very high confidence
2. 8\*8 frequency image D1

A screen shot of a computer

Description generated with high confidence

1. frequency image D1

A picture containing tennis

Description generated with high confidence

1. Frequency image D2

A screen shot of a computer

Description generated with high confidence

1. Images r1, r2

A screen shot of a person

Description generated with high confidence

A picture containing black, indoor, photo, white

Description generated with high confidence

Image R1 is more blurred than the image R2 because we have taken only the DC component for the image R1 and no other high frequency values this leads to the loss of data in the image

**Color based image segmentation**

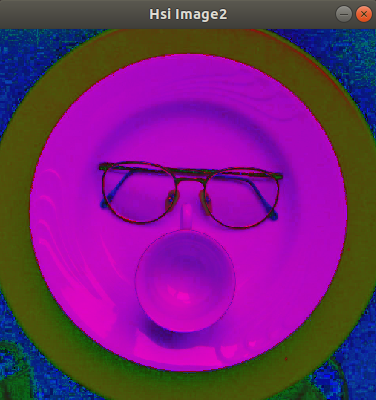
1. The original image are

A screen shot of a building

Description generated with high confidenceA picture containing plate, indoor, sitting

Description generated with very high confidence

1. The HSI image of the images are

A flat screen monitor

Description generated with high confidence

1. The canny edge images are

A close up of a logo

Description generated with very high confidenceA close up of a computer

Description generated with very high confidence

1. The region interest outline images are shown below and the lines are highlighted in blue color

A screen shot of a computer

Description generated with high confidenceA picture containing plate, indoor, sitting

Description generated with high confidence

**5. Bug Report**

* Did not implement the circle detection for the region of interest

**6. References:**

* <http://homepages.cae.wisc.edu/~ece554/website/Xilinx/app_notes/DCT_IDCT%20Customer%20Tutorial%20custdct.pdf>
* <http://answers.opencv.org/question/62446/conversion-from-rgb-to-hsi/>