CSCI 576 Homework 1

1.a Number of lines per frame = 450

Number of pixels per line = 520

Frame rate of camera = 25 Hz

Sub sampling scheme is 4:2:0 and each sample of Y,U and V is quantized with 8 bits
Thus average number of bits per pixel = (4*8)+(2*8)+(0*8)/4=12

Bit Rate = bits per sample*samples per sec*channels
= 450 * 520 * 12 * 25
= 70,200,000 = **70.2 Mbit/sec**

1.b Re-quantize chrominance signals (U and V) by 6 bits

Sub sampling scheme is 4:2:0 = (4*8)+(2*6)+(0*6) / 4 = 11

Minimum size of hard disk required to store 10 minutes of video = 450*520*11*25*10*60

= **38.61 Gbit**

2.a Given sequence = {1.8, 2.2, 2.2, 3.2, 3.3, 3.3, 2.5, 2.8, 2.8, 2.8, 1.5, 1.0, 1.2, 1.2, 1.8, 2.2, 2.2, 2.2, 1.9, 2.3, 1.2, 0.2, -1.2, -1.2, -1.7, -1.1, -2.2, -1.5, -1.5, -0.7, 0.1, 0.9 }

We need to quantize this signal by dividing the interval [-4,4] into 32 uniformly distributed levels. Thus 4-(-4)/32=8/32=0.25 (Round off by 0.25 value)

Quantized sequence = **{1.75, 2.25, 2.25, 3.25, 3.25, 3.25, 2.5, 2.75, 2.75, 2.75, 1.5, 1.0, 1.25, 1.75, 2.25, 2.25, 2.25, 2.0, 2.25, 1.25, 0.25, -1.25, -1.25, -1.75, -1.25, -2.25, -1.5, -1.5, -0.75, 0.0, 1.0}**

2.b There are 32 uniformly distributed levels (2^n=32) thus n=5

Total number of bits transmitted = number of distributed levels*bits per level

= 32*5 = **160 bits**

3.a Car is moving at speed 36km/hr

Number of frames per sec is 24

Diameter of tire = 0.4244 m

Distance covered in one rotation by wheel = pi * diameter

= 22/7 * 0.4244m

= 1.334m per rotation

Speed = 36 km/hr =36*1000/60*60 =10 m/s

Rate of tire rotation in rotations per sec = 10/1.333 = **7.5 rotations per sec**

3.b Filming rate of cam recorder = 8 fps

Rate of tire rotation is 7.5 rotations per sec

Number of degrees turned per frame = 7.5 * 360/8

= 337.5 degrees/frame

Therefore the wheel is covering 22.5 degrees per frame

22.5 degrees/frame * 8 fps/sec = 180 degrees/second

= 180/360 rotations/sec

= **½ rotations/second but in opposite direction**

3c. NTSC camera has R = 30 frames per second

Maximum frame per seconds possible is 30/2

=15 rotations per sec

Perimeter is 1.334m

Maximum speed at which the car can go is 15*1.334

= 20.01 m per sec

ANALYSIS QUESTIONS

1 Keeping U and V constant at 1 and varying Y we get the following distortion in images
We get the following output when Y=1, U=1 and V=1

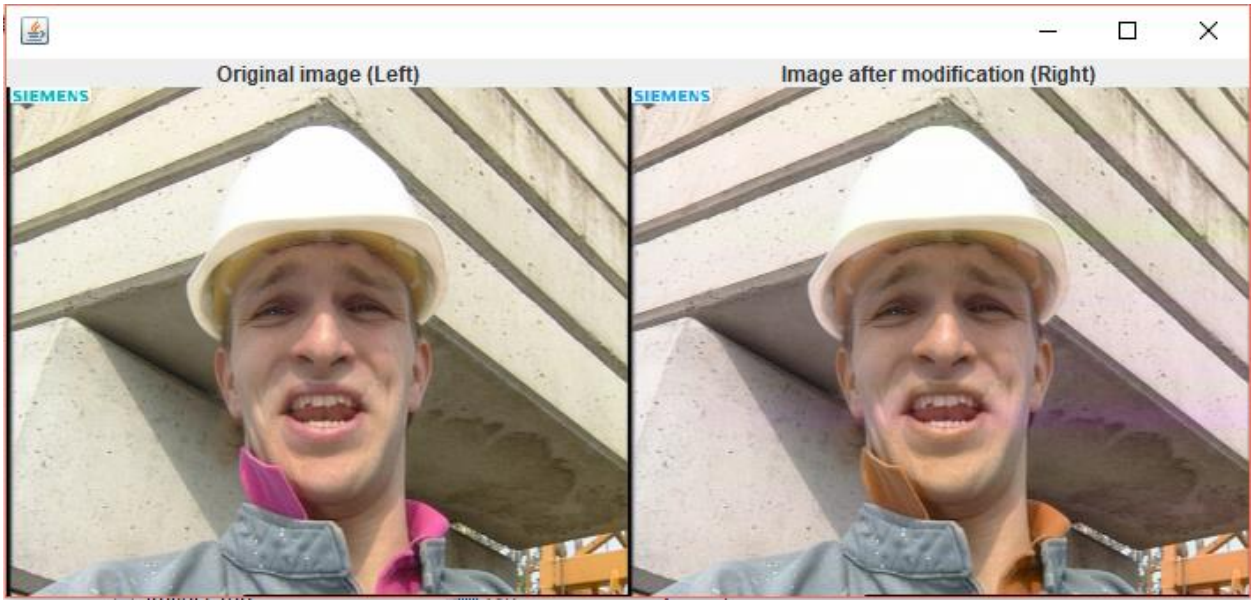
We start getting distortion in images when we alter Y value.

When we give Y=4, U=1 and V=1 we get the following output

When we give Y=8, U=1 and V=1, we get the following distortion in images

When we keep Y and V constant at 1 and change U values we get the following discoloration in image

Similarly when we keep Y and U constant at 1 and change V value we get the following image output



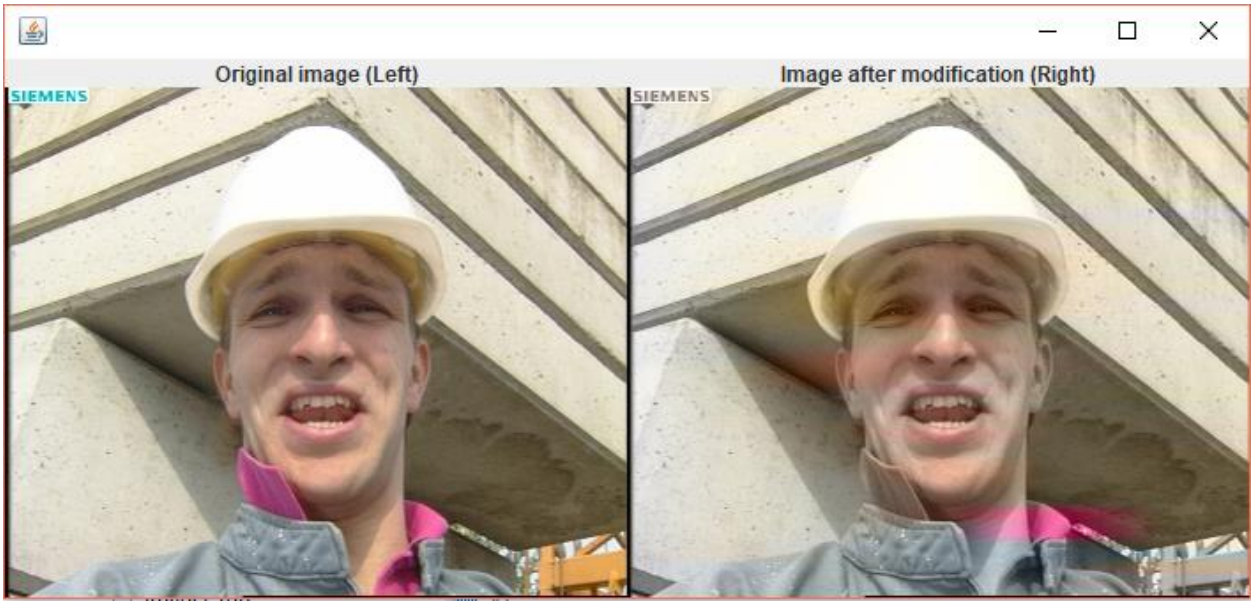
Thus from these changes we can infer that changing Y value majorly affects the image quality then changing the values of U and V

2. When we alter Y value the quality of image changes a lot but when we keep Y constant and change the values of U and V the quality of the image does not change drastically. When we keep Y constant at 1 and change U and V values there is a discoloration in images, this can be seen in the screenshots provided above.

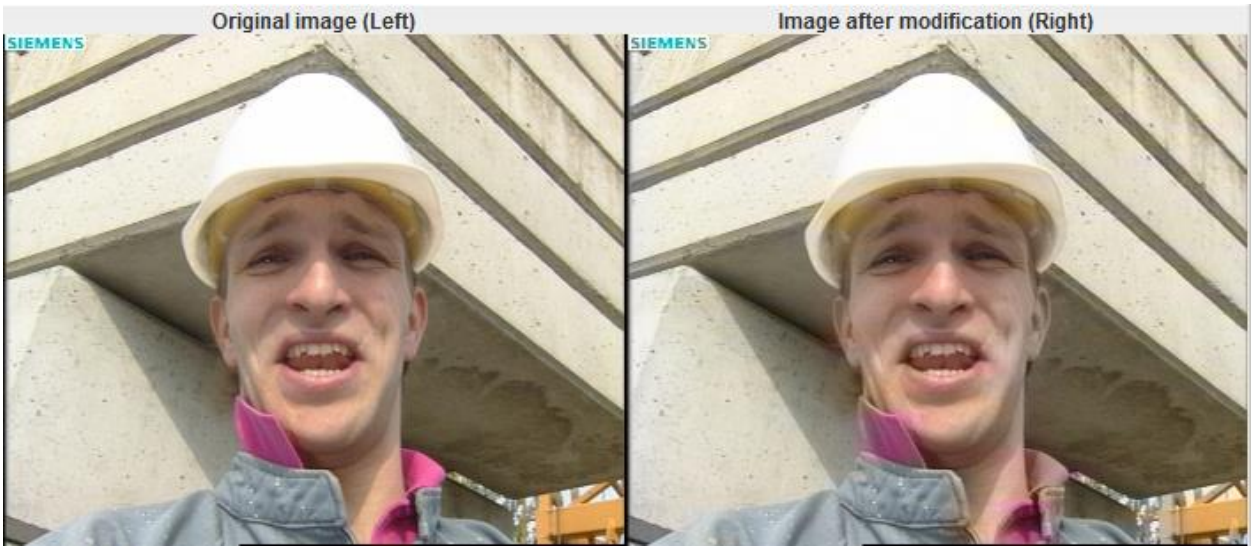
Thus to improve the quality of the image I have computed the **weighted average** value of the chrominance values (U and V) to fill the missing values of U and V, this helps in improving the image

quality drastically. I have also provided screenshots below before and after applying the weighted average mean method.

Before using weighted average method



After using weighted average method



3. I have used **Uniform Quantization Algorithm** where we break the color space into uniform cells. Find the cell that each color is in and map it to the center. This method performs poorly as it fails to capture the distribution of colors because some cells maybe empty.

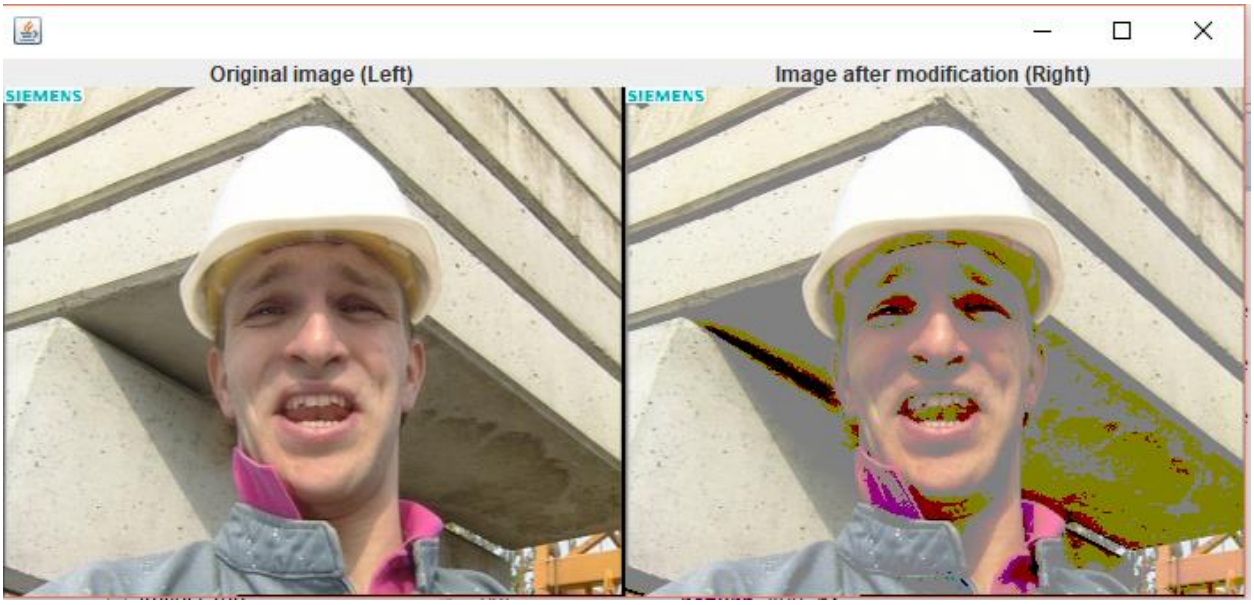
This method does very poorly on smooth gradients as lot of similar colors appear very close together.

Thus I propose to use the **Median Cut algorithm** where we look at distribution of colors and recursively

Find the “longest” dimension (r, g, b are dimensions).Choose the median of the long dimension as a color to use. Split along the median plane, and use recursion on both halves. This algorithm performs better. It uses kD-tree, a common form of spatial data structure.

Median cut works well as it divides up the color space in the most useful way

Output using Uniform Quantization



Output using Median Cut Algorithm