CSCI 576 Homework1

Tianxiang Tan

Q1. Suppose a camera has 450 lines per frame, 520 pixels per line, and 25 Hz frame rate. The color sub sampling scheme is 4:2:0, and the pixel aspect ratio is 16:9. The camera uses interlaced scanning, and each sample of Y, Cr, Cb is quantized with 8 bits

- What is the bit-rate produced by the camera?

  Since there are 450 lines and 520 pixels for a line, we can know that there are 450 * 520 = 234,000 pixels for a frame. And we also know that the frame rate is 25Hz, it means that the camera records 25 * 234,000 = 5,850,000 pixels for every second.

  For every pixel, since the sampling scheme is 4:2:0, we know that for every 4 pixel we will use 4 * 8 = 32bits for Y, 1 * 8 = 8bits for U and 1 * 8 = 8bits for V, that is 48bits in total and 48bits/4pixels=12bits/pixel on average.

  Therefore, the bit rate for the camera is 12 * 5,850,000 = 70,200,000 bits/second.

- Suppose we want to store the video signal on a hard disk, and, in order to save space, re-quantize each chrominance (Cr, Cb) signals with only 6 bits per sample. What is the minimum size of the hard disk required to store 10 minutes of video?

  Since we only change the bits for per sample, the frame rate is unchanged and the camera still needs to record 5,850,000 pixels per second.

  However, since we use 6 bits for Cr, Cb now, so for every 4 pixels, we will use 4 * 8 = 32 bits for Y, 1 * 6 = 6bits for U and 1 * 6 = 6bits for V, that is 44 bits in total and 44bits/4pixels = 11bits/pixel on average.

  Therefore, the bit rate for the camera now is 11 * 5,850,000 = 64,350,000 bits/second. And 10minutes = 60seconds * 10 = 600seconds. So we record 64,350,000bits/second * 600seconds = 38,610,000,000bits in total.

  That is 4,826,250,000bytes that is 4.5GB approximately (divided by 1024).

Q2. The following sequence of real numbers has been obtained sampling an audio signal: 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 Quantize this sequence by dividing the interval [-4, 4] into 32 uniformly distributed levels (place the level 0 at -3.75, the level 1 at -3.5, and so on. This should simplify your calculations).

- Write down the quantized sequence.

Original sample signal Sequence	Quantized sample signal Sequence
1.8	1.75
2.2	2.25
2.2	2.25
3.2	3.25
3.3	3.25
3.3	3.25
2.5	2.50
2.8	2.75
2.8	2.75
2.8	2.75
1.5	1.50
1.0	1.00
1.2	1.25
1.2	1.25
1.8	1.75
2.2	2.25
2.2	2.25
2.2	2.25
1.9	2.00
2.3	2.25
1.2	1.25
0.2	0.25
-1.2	-1.25
-1.2	-1.25
-1.7	-1.75

-1.1	-1.00
-2.2	-2.25
-1.5	-1.50
-1.5	-1.50
-0.7	-0.75
0.1	0.00

- How many bits do you need to transmit it?

Since we need 5bits to represent a number and there are 32 numbers here, so it means that we need 5bits to transmit a number and we need 32 * 5 = 160bits to transmit the sequence of the numbers in the problem.

Q3. Temporal aliasing can be observed when you attempt to record a rotating wheel with a video camera. In this problem, you will analyze such effects. Assume there is a car moving at 36 km/hr and you record the car using a film, which traditionally record at 24 frames per second. The tires have a diameter of 0.4244 meters. Each tire has a white mark to gauge the speed of rotation.

- If you are watching this projected movie in a theatre, what do you perceive the rate of tire rotation to be in rotations/sec?

  We know that 36km/hr = 10m/s and we take 3.14 for π. So the perimeter for the tire is about π * 0.4244m = 1.66577m. Therefore, for every second the tire rotates 10 / 1.66577 = 7.5 rotation/second approximately. Since the frame rate for the film is 24 which is larger than 2 * 7.5, so from Nyquist's sampling theorem we know that we can "perfectly" see the tire rotates 7.5 circles per second in the same direction, which it rotates in the real world, when we record it.

- If you use your camcorder to record the movie in the theater and your camcorder is recording at half the film rate (ie 12 fps), at what rate (rotations/sec) does the tire rotate in your video recording?

  Suppose the white mark is at the top of the tire and the wheel is rotating clockwise. Since the camcorder records the movie in theater (24fps), so it means that the camcorder will capture the 2nd frame recorded by the film at 1/12 second, and 4th frame by the film at 2/12 second, so on and so forth. This means that the camcorder recording the movement of the wheel in the theater is the same as it recording the wheel in the real world, since what it capture is the wheel exactly at 1/12 second, 2/12 second so on and so forth. So for every 1/12 second, the wheel will rotate 7.5*2 π /12 = 5/4 π in clockwise. If we build a coordinate which denotes the top of the tire as 0 π and the value increase clockwise, so for the time 0s, 1/12s, 2/12s the white mark is at 0 π , 5/4 π , 1/2 π in the coordinate respectively. And we can draw the white mark position for every 1/12 second in this way. After that, we can see that the wheel will rotate 3/4 π counter-clockwise every 1/12 second in the camcorder, that means the wheel will rotate 9 π in a second. Its speed is 4.5 circle/second and its direction is couter-clockwise.

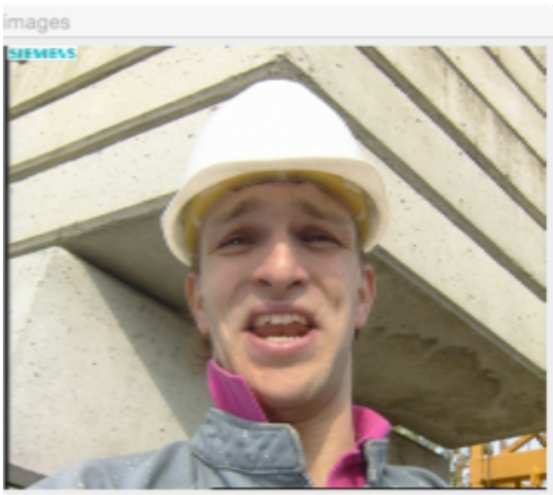
Programming Part

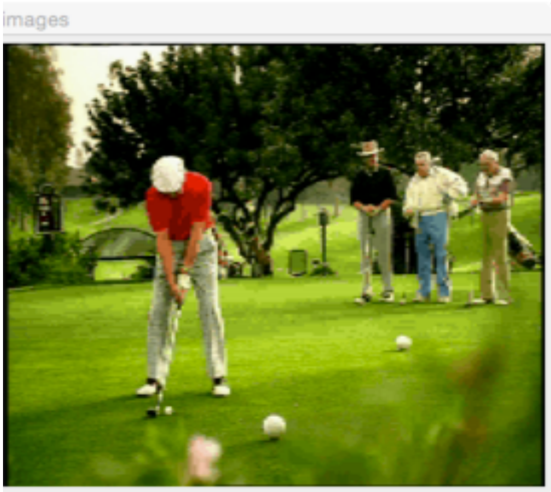
Analysis 1

We first keep the U, V unchanged and change the Y from 2 to 4, the result is showed below (the left is the original image, the right one is subsampled image):

We first keep U and V 1 and 1 respectively and vary Y.

Y U V: 2 1 1





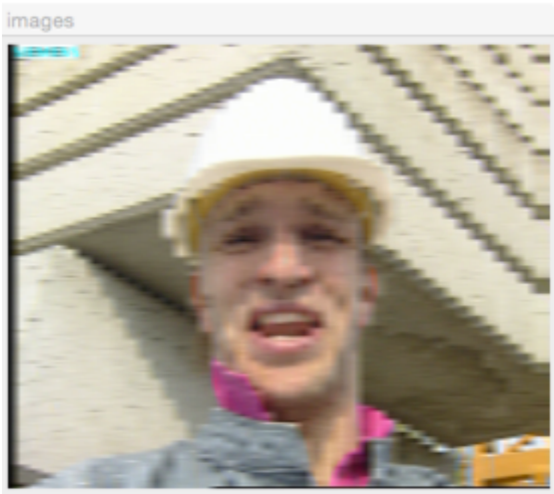
YUV: 4 1 1

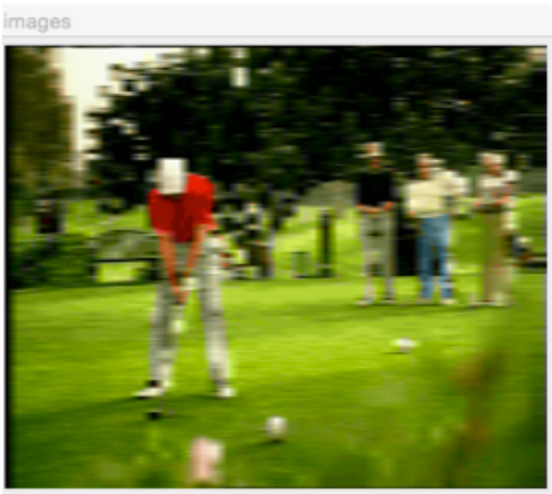
images

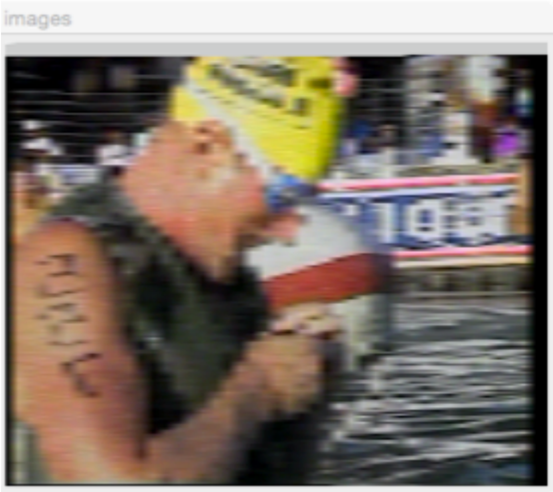
images

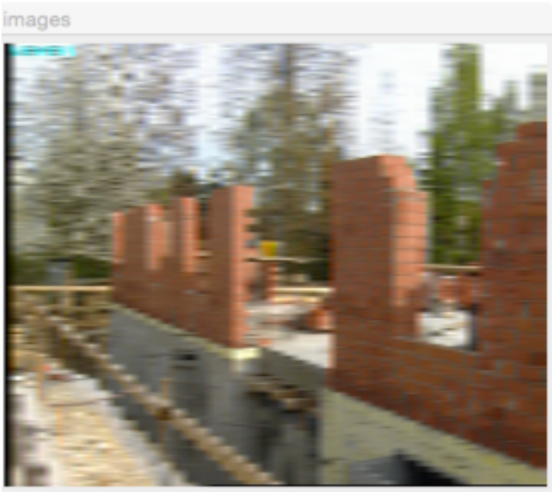
images

YUV: 8 1 1



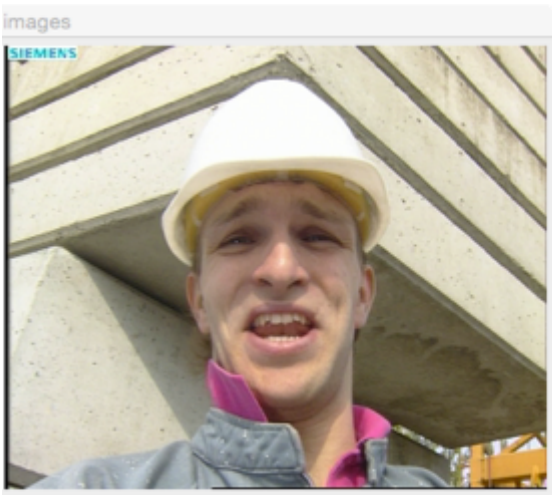


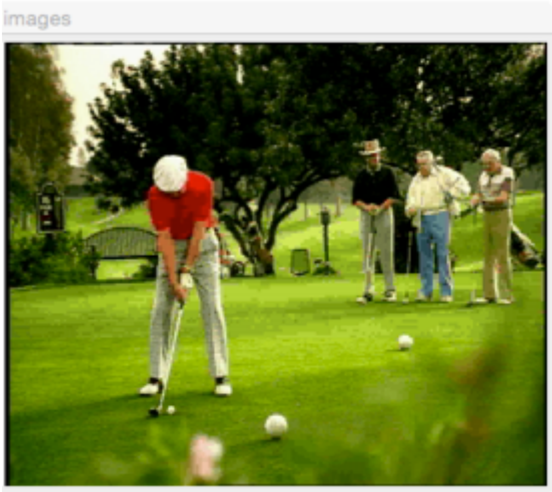




And then we keep Y and U 1 and 1, and vary V

YUV: 1 1 4





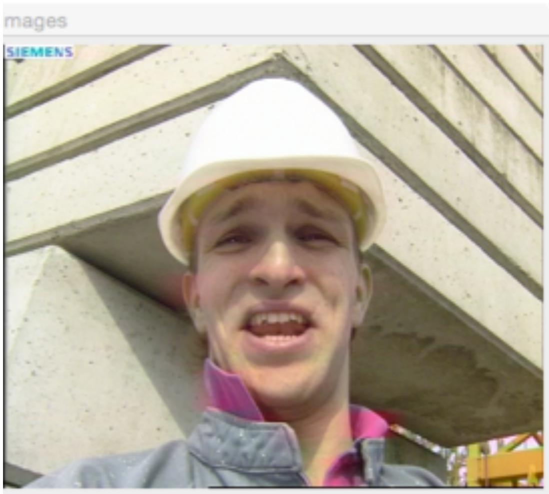
Y U V: 1 1 16

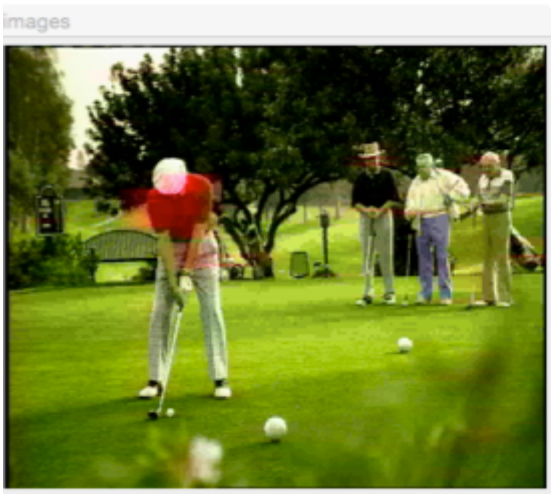
images

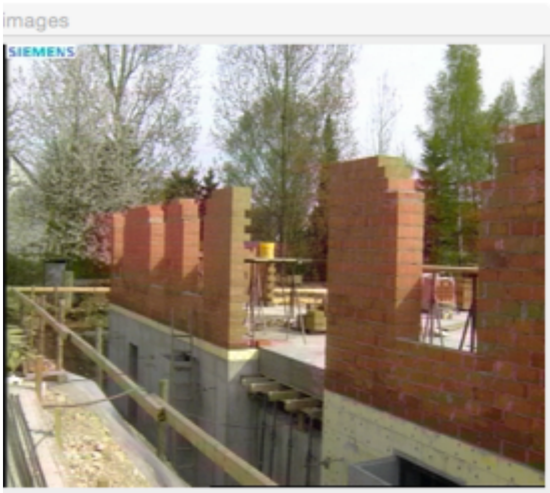
images

images

Y U V: 1 1 24

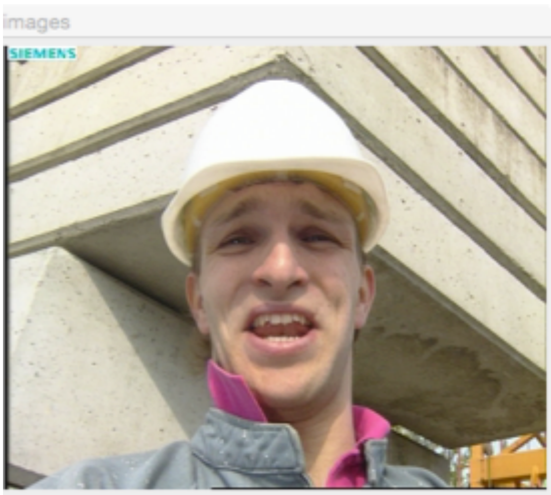


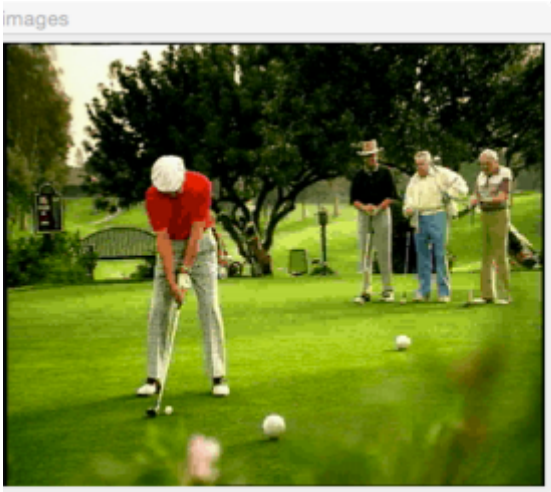




In the end, we keep Y V at 1 and 1, and vary U:

Y U V: 1 4 1





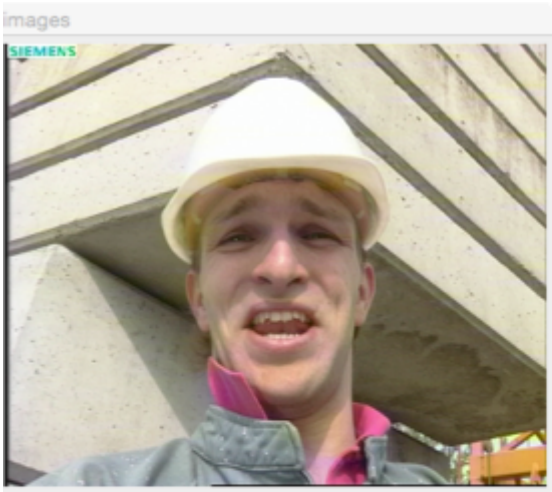
Y U V: 1 16 1

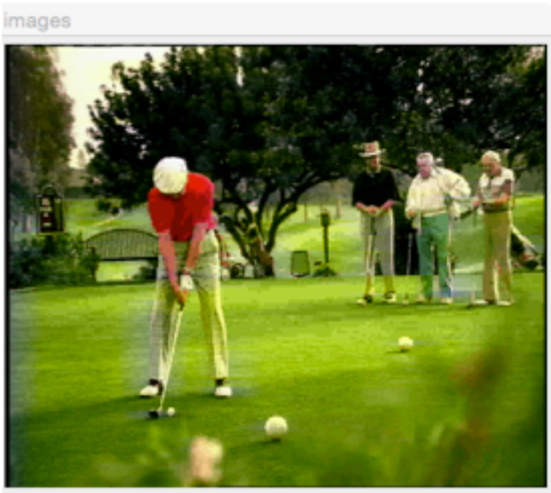
images

images

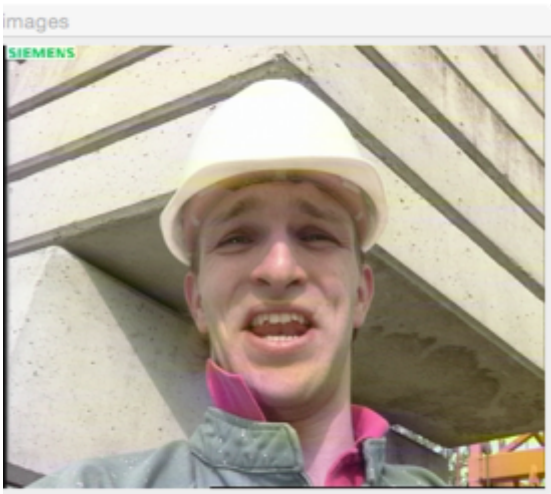
images

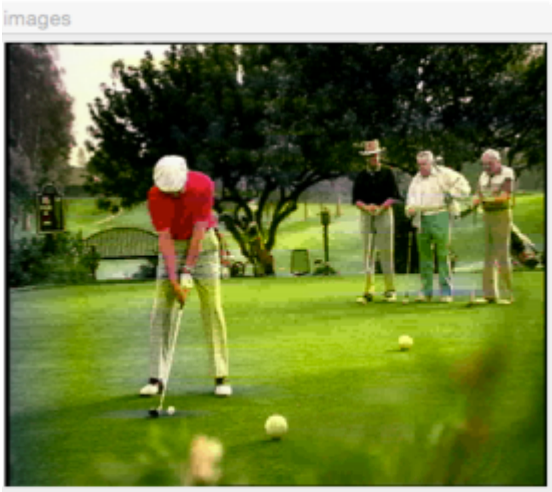
Y U V: 1 24 1





Y U V: 1 48 1





Summary

(My feeling to the images. 3 present feeling good, 2 for so so, 1 for bad)

YUV Subsample	Image1	Image2	Image3	Image4
2 1 1	3	2	3	3

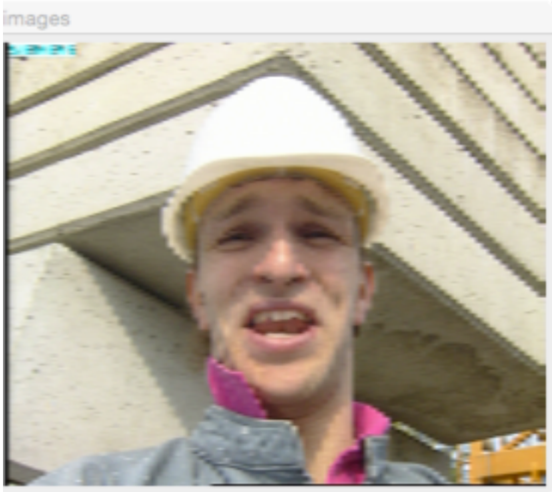
4 1 1	1	1	1	2
8 1 1	1	1	1	1
1 1 4	3	3	3	3
1 1 16	3	1	3	2
1 1 24	2	1	2	2
1 4 1	3	3	3	3
1 16 1	3	2	3	3
1 24 1	3	2	3	2
1 48 1	2	2	2	2

Analysis: From the result images, we can see that even though we subsample the Y channel with a very small value 4, I feel that the image get blurred after subsampling. However, for the U and V channels, when we subsample them with value 16, which is 4 times larger than the Y channel, I still feel it is the same image after subsampling. Only when the value goes up to 24, I will feel the some of the colors in the image change. Moreover, the subsampling for UV affects each image independently. We can see that in image 4, most of the pixels are red and green. So the loss of UV channel data will affect the feeling to the image. And I give it lower rate compare to most of the other images with the same subsampling parameters. Besides for image2, most of the pixels are green and our focus is on the people in red. So, the loss of UV channel data will also affect my feeling greatly, so lower rate is given compared to other images. For images 1 and 3, since they are mainly white and black, which does mostly the Y channel affect, so the loss of UV channel doesn't have a great impact to the feeling of the image.

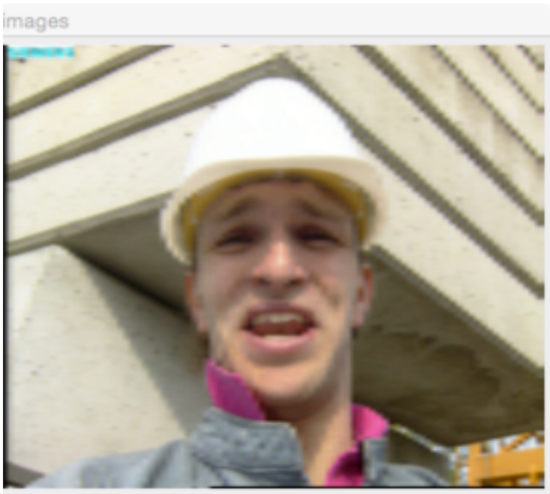
So my conclusion is that the Y channel has a larger effect on the image than the UV channels, we need to keep as much as Y channel data possible to make the subsampled image looks as the same as the original one. And UV channels have different impacts on different images with different main colors.

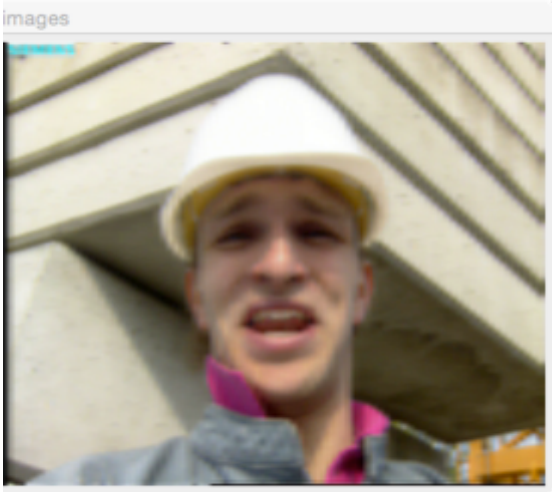
Question 2

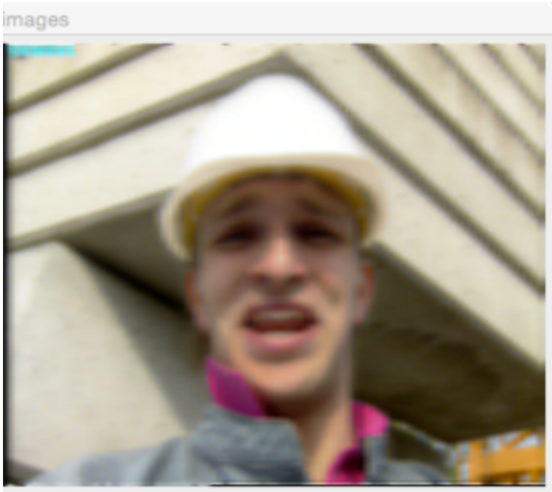
Let's use the YUV 4:1:1 image for the discussion(show below).



We can see that there are many small triangles after subsampling in the image. This is caused by the loss of Y channel data. To reduce the effect of aliasing, we need to reduce the frequency of the original image before subsampling it. One way to do that is to use an average filter and do a preprocess before subsampling. Here, for the YUV 4:1:1, there are 3 average filters with size 3*3, 5*5, 7*7, the result is shown below respectively:





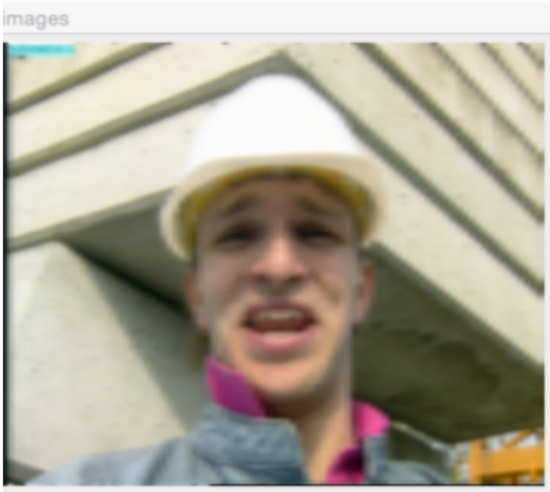


(the first one is 3*3, the second is 5*5, the last one is 7*7).

So we can see that the last two images we can the aliasing is disappeared, but the image get blurred.

The process can be done before we subsample the image. Typically, we can do it in the RGB color space before transferring into YUV space. But we can also only filter the Y channel after the transferring to get the same result. (I don’t filter the UV channels, since they record the color info, and we will get weird result if we filter them) I prefer do it in the YUV color space, since the color looks more close to the original image. (see below)

Filtering in RGB color space (5*5 average filter):



Filtering in YUV color space (5*5 average filter):