

CS 576 Spring 2019 – Assignment 1

Instructor: Parag Havaladar

Assigned on Monday Jan 14 2019

Solutions due Monday Feb 4 2019 (by 12:00pm *afternoon*)

Late Policy: None, unless prior arrangement has been made for a valid reason.

Written Part: (20 points)

Each question has marks displayed

Q.1 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? (2 points)
- Suppose we want to store the video signal on a hard disk, and, in order to save space, the signal is re-quantized so that each channel (Y, Cr, Cb) uses 6 bits . What is the minimum size of the hard disk required to store 10 minutes of video (3 points)

Q.2 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. (4 points)
- How many bits do you need to transmit it? (1 points)

Q.3 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? (2 points)
- If you use your camcorder to record the movie in the theater and your camcorder is recording at one third film rate (ie 8 fps), at what rate (rotations/sec) does the tire rotate in your video recording (5 points)
- If you use an NTSC camera with 30 fps, what is the maximum speed that the car can go at so that you see no aliasing in the recording (3 points)

Programming Part: (130 points)

This assignment will help you gain a practical understanding of issues that relate to video resampling, spatial and temporal aliasing effects, image aspect ratios and pixel aspect ratios. In this assignment you will be given a video file as input and produce a spatially & temporally resampled output.

We have provided a Microsoft Visual C++ project as well as a java project to display an image. This source has been provided as a reference for students who may not know how to read and display images. You are free to use this as a start, or write your own in C/C++ or any other programming language such as Java. For assignments, we do not allow scripting environments such as matlab, python.

Input to your program will be 6 parameters where:

- The first parameter is the name of the input video file. (file format description provided)
- The second parameter is a floating pointing number that will be the scaling factor for your width.
- The third parameter is a floating pointing number that will be the scaling factor for your height
- The fourth parameter controls the output frame rate (that will change your video playback)
- The fifth parameter controls whether anti-aliasing should be turned on. A value of 0 indicates this should be off, and a value of 1 indicates it should be switched on.
- The last parameter controls the analysis/extra credit portion of your assignment – could have a value of 0, 1, 2 or an optional part 3, where 0 represents the default implementation, 1 represents analysis part A, 2 represents analysis part B, and 3 represents the (optional) extra credit part1.

Data Format:

Remember all input files for this assignment will have the same format as explained to the class website. They will be of size quarter HD 960x540 (CIF format), at 10 fps. The format of your native input video will consist of a list of pixels frame-by-frame, channel-by-channel with every pixel represented as three bytes (or chars), one byte (or char) per channel. To elaborate, the data will be stored as *frames* - frame1, frame2, frame n. Each *frame* is further stored as rows of bytes for each channel - *rrr...bbb...ggg....* And each *r*, *g* and *b* is a byte (or char) long.

Typical invocations to your program would look like.

MyExe Video.rgb 1.0 1.0 10 0 0

This should not change you input and the output is the same as the original file

MyExe Video.rgb 0.5 0.5 10 0 0

This should change your video file to half its size at the same frame rate

MyExe Video.rgb 0.5 0.5 10 1 0

This should change your video file to half its size at the same frame rate but have anti aliasing turned on.

MyExe Video.rgb 1.0 1.0 5 0 0

No change in spatial, but video should play at 5 frames a second (which is half as fast and the video plays twice as long).

Implementation (70 points)

Your implementation should create a video player that takes the described parameter as input, scales the images and appropriately display the video at the right frame rate. You will need to extend the given code (or write on your own) to provide

- Functionality to read the whole video file
- Ability to resize each frame
- Implementation of an averaging filter to perform anti aliasing as discussed in class. Here instead of copying the pixel value (r,g,b) from your source position to the destination position, you copy the average of a small 3x3 window from your source position to your destination position.
- Render video that plays the resized output video in a synchronized manner as per the fps parameters

Analysis Part A – dealing with Pixel Aspect Ratio (30 points)

The original image aspect ratio of the video is 1.777:1 and the pixel content are square (nothing appears stretched). The output has unstretched pixels as long as both the width and height scaling factors are the same. When the width and heights are scaled non-proportionally, there is a pixel scaling or pixel stretching effect that changes your pixel aspect ratio. This is disturbing, and not desirable, but unavoidable in these cases. There are simple smart operations that can minimize this effect. One suboptimal solution is letter boxing, where the black areas are minimally fitted to the top/bottom or left/right sections to maintain the same pixel aspect, although standardly used, this does not make good use of your screen real estate. One smart improvement to this would be non-linear mapping, where your mapping changes according to where the pixels are on the screen. So while the target width and height are maintained, the mapping is controlled non linearly such that a larger part of the central area (which is the focus of attention) has the same original pixel aspect, but the peripheral areas seem scaled to compensate.

Implement a nonlinear mapping function that appropriately changes the content.

This will be invoked by using 1 for your last parameter.

Analysis Part B – dealing with Pixel Aspect Ratio (30 points)

While part A deals with changing pixel aspects, there have been some smart “content aware” rescaling functions which have made their way to commercial applications. As demonstrated in class, recent work by Ariel Shamir from Mitsubishi Electric Research Lab (MERL) shows how to intelligently resize images by being *content aware* by a gradient domain-based energy minimization process called “*seam carving*”. See example below



Original Image



Scaled



Cropped



Seam Carved

Go to (<http://www.faculty.idc.ac.il/arik/SCWeb/imret/index.html>) and understand how the algorithm works by reading the paper and examples.

Various source code is available on various platforms (code.google.com or github)

You may reuse source code for this algorithm to implement content aware rescaling in width and height. You will need to apply the algorithm to each frame independently to resize the video.

Your program will be invoked using the last parameter as 2.

Optional Extra Credit – temporal content aware remapping

The paper talks about seam carving on a single image. Extrapolating from the paper and/or observing your results you will see temporal inconsistency when this method is applied to each frame individually. This is because content aware remapping deals with each frame and not a stack of frames simultaneously. Think about what changes you need to make to minimize temporal inconsistency. Implement your version to show this.

This optional part, if implemented, will be invoked by using the last parameter as 3

What should you submit ?

- Your source code, and your project file or makefile. Please confirm submission procedure from the TAs. **Please do not submit any binaries or data sets.** We will compile your program and execute our tests accordingly.
- Along with the program, also submit an electronic document (word, pdf etc) for the written part. Also mention therein whether you have attempted extra credit so that we can evaluate your submission accordingly.

Grading expectations:

For written sections, you will be awarded full points for all correct answers and perhaps partial points depending on the question. For the programming part we will be conducting tests to see the output and evaluate your results based on (among other things) correctness of output scaled size, frame rate, whether aliasing has worked. Etc. If you have done extra credit please let us know in your written submission part explicitly so that we can evaluate it.