MATH 2221 Mathematics Laboratory II

Programming Homework

Name: ________________________ Student ID.: ________________________

Instructions

1. To test whether you write and understand your own program, you are required to hand in your homework in person on or before the due date. The TA will ask several questions about your program. You must answer the questions satisfactorily in order to get marks. If you cannot answer them, you will not be given any marks and your name will be given to the teacher for further investigation.

2. Due date: 15 May 2017. Late homework will NOT be graded, and will receive a score of zero. Please hand in your homework as early as possible so you don't have to wait in line for a long time on the due date. You can hand in your homework in person to the TAs in LSB222B any time on or before the due date. For convenience, it's better to make an appointment with the TAs if you want to hand in your homework before the due date. Please make an appointment through the following Email address:

   `kckchan@math.cuhk.edu.hk`

3. Please bring a hard copy of your program as well as a soft copy of your program on a USB.

4. Please do not let others copy your programs or results as we have no way to tell who is copying from whom and you may be liable for the penalties.

5. To check our updated news, please visit the link below:

   `http://www.math.cuhk.edu.hk/course/1617/math2221ab`

I declare that the assignment here submitted is original except for source material explicitly acknowledged. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website
http://www.cuhk.edu.hk/policy/academichonesty/

Signature	Date

An introduction to digital images and videos

Gray-scale images are images with different levels of grayness in each pixel of the image. Some of the pixels may be completely black, some may be completely white and some have grayness in between. The grayness of a pixel is called the intensity of the pixel or pixel value.

In MATLAB, gray-scale images are represented by two-dimensional arrays (matrices). Each element of the matrix corresponds to a single pixel in the image. For example, an image with 200-by-300 pixels is stored as a 200-by-300 matrix. The value of the entry is precisely the pixel value (or intensity) at that particular pixel.

This convention makes working with grey-scale images similar to working with matrices. For example, you can select a single pixel from an image using normal matrix subscript such as

$$\mathtt{I(2,15)}$$

This command returns the value of the pixel (or the intensity of the pixel) at row 2, column 15 of the image I.

Usually, there are 256 levels of grayness with 0 representing completely black and 255 representing completely white. Different levels of grayness (or intensity) will be represented by different values in between 0 and 255. Thus the pixel values can be represented by 8-bit unsigned integers. For an intensity image whose image values are double-precision floating point, 0 is black and 1 is white. For an intensity image whose image values are 16-bit signed integers, -32768 is black and 32767 is white. Figure 1 shows how an array of 8-bit unsigned integers represents an image.

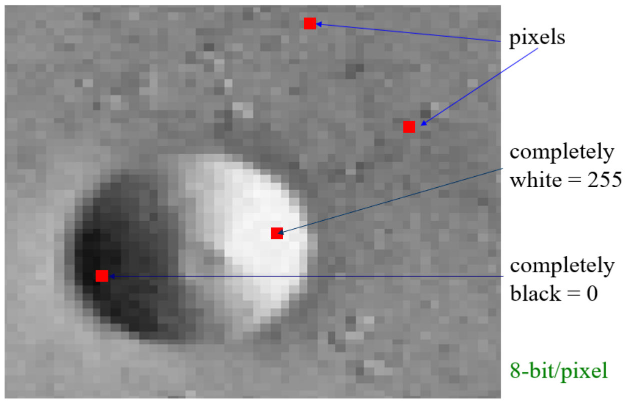

Figure 1: An image in MATLAB

Color images are usually stored in the RGB (red-green-blue) format where each channel (R, G, B) represents the intensity of the pixel in that channel. Thus a color image of m-by-n pixels requires a three-dimensional array (an m-by-n-by-3 array) to represent it. The $(:,:,1)$ coordinate plane represents the red pixel intensities, the $(:,:,2)$ the green pixel intensities, and $(:,:,3)$ the blue pixel intensities.

A video is just a sequence of images (called frames). Usually there are 10 to 30 images per second. In Matlab, videos are read frame by frame into the computer memory. Gray-scale videos are sequences of gray-scale images where each frame is presented by a matrix. Videos in RGB format are sequences of color images where each frame is represented by a 3-dimensional array.

Part I (60 marks)

The horizontal and vertical resolution of a video is related to the number of columns and number of rows in each frame. MATLAB provides tools for video processing, such as functions and system objects that read and write video files, etc. Following the instructions below to transform the low-resolution video "cargray.avi" into a high-resolution one using bilinear interpolation. The video can be downloaded at:

`www.math.cuhk.edu.hk/course_builder/1617/math2221ab/SourceVideos.zip`

1. Read the gray-scale video from the file "cargray.avi" and extract the frames (images) from the video. To see how images are stored in MATLAB, search Image Types in MATLAB help browser.
2. Compute the number of frames f in the video, and the number of rows n and number of columns m of the first frame.
3. For each frame, use bilinear interpolation to create a new image with size $(2n-1)\times(2m-1)$. For the interpolation, please write your own code (i.e., you are not allowed to use built-in function like imresize, interp2 etc).
4. Write the new images into a video named "newcargray.avi" in the format of "uncompressed AVI file with gray-scale". Set the frame rate as 16 frames per second.

Part II (40 marks)

The video for this part can be downloaded at:

`www.math.cuhk.edu.hk/course_builder/1617/math2221ab/SourceVideos.zip`

1. Do the followings to create a high-resolution color video.
   (a) Read the color video from the file "Book.avi" and extract the frames (images) from the video.
   (b) Compute the number of frames f in the video, and the number of rows n and number of columns m of the first frame.
   (c) For each color channel (i.e., red, green, blue) in each frame, use bicubic interpolation to create a new one with $2n \times 2m$ pixels. You may implement the bicubic interpolation yourself or use MATLAB function imresize. Store the data you get in a $2n \times 2m \times 3 \times f$ array.
   (d) Play the video using the data in last step with MATLAB.
   (e) Write the data into a video named "newBook.mp4" in the format of "MPEG-4 file". Set the frame rate as 30 frames per second.

2. (Optional) Can you use another method (either interpolation or non-interpolation) to create a new video with "Book.avi" such that
   (a) Each image in the video has $2n \times 2m$ pixels.
   (b) The video could be as clear as possible.