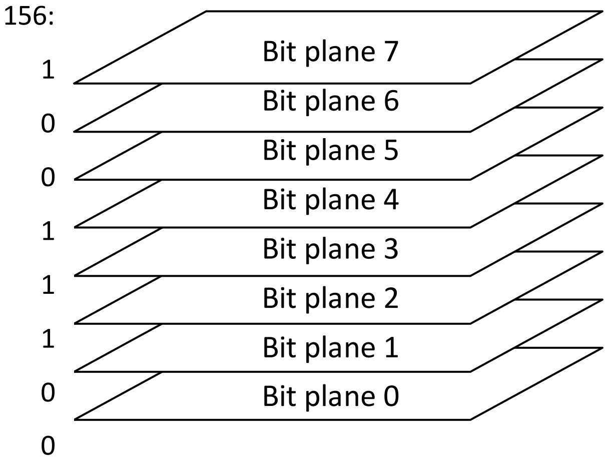
ENGI 4559 Digital Signal and Image Processing – Fall 2015
Project 3: Image Enhancement (15%)

Part 1: Bit-Plane Slicing (5%)

Bit-plane slicing is an image enhancement technique we have not discussed in class. In this method, the image is divided in a set of 1-bit planes, one for each bit of the pixels' intensity bits. Each plane is thus a binary version of the image, with each pixel having a 1 or 0 depending on that bit in the intensity value of the pixel in the original image.

For example, since the images in our class use 8-bit intensity values, the image will be divided in eight planes. A pixel of intensity 156 (10011100) will have a 1 in planes 2, 3, 4, and 7, and 0 on planes 0, 1, 5 and 6.



Implement a bit-plane slicing algorithm in Matlab. Download the "lenna.png" image from the course website. Experiment with the "Lenna" image; display each bit plane individually, as well as reconstructions of the image using only some of the bit planes. Discuss your results. What does each bit plane's image look like, and what do images reconstructed from some of the planes look like? How do you explain this? Which of the image processing applications we have learned about in class could this decomposition be useful for?

Hint: Matlab has a function "`dec2bin`" which converts a decimal number to a string that represents the binary number.

Intermission: Background Subtraction

An application of image processing that we have not discussed in class is to automatically detect changes between two similar images. We refer to the changes as the "action" or "foreground", and the part that remains constant as the "background". This is notably a common task in video processing, to detect moving objects in the sequence of images. The goal then becomes to generate a new sequence of images featuring only the foreground and excluding the background.

Multiple techniques for background subtraction exist. A simple one is based on the assumption that the background is both known and static throughout the sequence of images (or that it changes very little compared to the object in action). In that case, we can simply subtract the known background from the images. The background visible in each image will be reduced to zero (or near zero if small changes are present), while the object will be clearly apparent as a large value in the subtracted image (i.e. it is a large variation compared to the known background). We can then return to the original images and keep only those pixels that show a large value in the subtracted image, which we know correspond to the object in action.

Part 2: CCTV Image Processing (10%)

Given the absolutely amazing training you received in digital signal and image processing, you managed to obtain a job in London managing the CCTV camera network. On your first day at your new job, you have to handle a camera watching over a very, very important back alley. An image of the back alley seen by the camera on a typical clear sunny day is given in “backalley.jpg”, available on the website. Unfortunately, the camera is rather bad and the weather conditions are very variable, so the image quality can fluctuate a lot. Moreover, criminals tend to come in not on clear sunny days but during weather extremes, when the camera is at its least reliable.

A set of 16 bad-weather images taken by the camera has been provided to you in the “backalley.zip” file available on the website. Your task is to design a Matlab program to analyse them and automatically determine which ones feature intruders.

Describe your program in detail. Describe every image processing step, explain what it accomplishes and justify its use in your program. If your program uses constants or fixed threshold values, state their values and explain how you picked them.

Hint: To help get you started, the code “`Backalley.m`” available on the website contains a simple loop that reads and displays each backalley image one after the other.

Submission and evaluation:

Print out a report in which you present your results and your discussion on the questions asked in this project, and hand it to the professor in class on November 28th. Do not hand in your Matlab code. This project must be done individually. Do not forget to write your name on your report.

For each part, half the marks will be given for demonstrating you did the work correctly, and half the marks will be given for a discussion demonstrating that you understand the work you did. Bonus marks (up to a maximum final grade of 100%) will be given for innovative and justified additional experiments and analysis. Penalty marks (up to 10% of a part value, rounding up) will be deducted for poor writing, spelling, or formatting of the report.