STONY BROOK UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE 527: INTRODUCTION TO COMPUTER VISION

Handed out: January 31, 2008 **Due on:** Never

Problem Set #0

Problem 1: Image correction for an image digitizer.

Load in and display the image original.tif. This is the (synthesized) output from a hypothetical 35mm slide scanner. The slide to be digitized is illuminated with a lamp and imaged onto a plane in space. A 1-d detector array is moved through that image plane and records the amount of light falling on each pixel as it scans through the image. original.tif is the raw output of a scan. Each vertical column represents the output of a single pixel of the detector. Each horizontal row represents a different time step of the detector through the image.

Notice the vertical streaks through the image. These are because not every element of the detector array has equal sensitivity to light. Since the 1-d detector array sweeps through the image vertically, the less sensitive array elements leave dark streaks through the image. You remove the slide from the scanner, and record a calibration "white field" image (whiteField.tif). This image shows the vertical streaks, and also unevenness in the illumination of the slide area. Both these multiplicative effects can be compensated for by dividing the slide image by the white field image (point-by-point division, using "./", not the matrix division operator, "/"). In other words, at every pixel position, divide the intensity of each color of the original image by the intensity of the white field image at that position. That will normalize for the differences in sensitivity and illumination. Display the resulting corrected image.

Solid-state detectors typically have a linear response to light intensity, while monitor displays typically have a non-linear response, often modeled by an exponential non-linearity. Find a compensating nonlinearity which leads to a pleasing tonescale for the displayed image by raising each pixel to some power in the range of 0.3 to 0.5. Be aware of the overall image scale.

Problem 2: CCD color interpolation.
To record color images with the single-chip CCD cameras typically used in digital cameras, arrays of color filters are used. Each pixel is covered by a filter of some color, giving a sample of only one color at each spatial position on the detector (see slides in ccd.ppt). Digital cameras need to reconstruct an image with 3 color samples at each position from the recorded data of just one color sample per position. (The human eye needs to do this, too.) Here we'll show one simple interpolation method, to practice image array manipulations in matlab. Load in the 3-color image brettan.tif.

(a) Simulate the sampling by the CCD. Assume that the color filter pattern is R, G, B stripes, in a repeating pattern (as shown in
ccd.ppt). Display a black and white image showing the intensity of each pixel in the color band of the color filter stripe that it lies under.

(b) Linear interpolation. Form a color image from the simulated data of part (a) by linearly interpolating between nearby color samples in the same row. For example, if the pixel at position (1,1) is a red sample, then the red value at position (1,2) should be

$$I(1, 2) = \tfrac{2}{3} I(1, 1) + \tfrac{1}{3} I(1, 4),$$

where *I* is the array of sampled intensities from (a). Display the resulting full-color picture.

(c) The undersampling causes color fringe artifacts. To reduce those, the image can be optically blurred before falling on the CCD. Load in the blurred image, brettanBlurred.tif, and perform the same steps as in (a) and (b). Are the color fringes reduced? At what price? Later in the term, we'll discuss a non-linear color sample interpolation method that requires much less image blurring in order to remove color fringe artifacts.