HCI/ComS 575X: Computational Perception
Homework 1, Part II (Basic Matlab)
Out: Wed. Jan 16, 2008
Due: Mon. Jan 30, 2008
Submit this part in electronic form on your section of the class wiki.

This part requires you to use matlab to perform some simple imaging and plotting operations.
Please refer to the matlab tutorial from last week (or google) if you have any problems.

1. **Image Transformation**

   Start with a color JPEG image of yourself. Load it in matlab. Convert it to grayscale. Flip it so that the image is upside-down. Finally, resize it to be 1/2 of the original size (or 1/4 of the original area). Save the resulting image. Post your original image, your result image, and your matlab code on your wiki page.

2. **Try Some Other Functions**

   Pick your favorite function from matlab's image processing toolkit. Convert the image that you used in 1) to grayscale and perform the function on the image. Now choose two more functions and do the same. Post your original image, your three resulting images, and your code on the wiki.

3. Plotting Different Functions

   a) Write a short matlab program that plots the probability density function of a 1D Gaussian function with mean $\mu = 1.5$ and standard deviation $\sigma = 2.0$.

   b) Write another matlab program that plots the probability density function of a 2D Gaussian function with mean $\mu = (\mu_x, \mu_y) = (1, -1)$ and a diagonal covariance matrix Σ as shown below:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

   where $\sigma_x^2 = 1$ and $\sigma_y^2 = 4$. Hint: In this case the result should be a surface.

   c) Plot the ellipse given by the equation $\frac{(x-x_0)^2}{a^2} + \frac{(y-y_o)^2}{b^2} = 1$. It is up tp you to choose values for x_0, y_0, a, and b.

   d) Plot the cos(x) function in the interval $[-2\pi, 2\pi]$.

   e) Plot the ellipsoid specified by the equation $\frac{x^2}{1} + \frac{y^2}{4} + \frac{z^2}{9} = 1$ in 3D.

   When you are done post your five programs and your resulting plots (matlab can save them for you; check the figure window menu) on the wiki.

HCI/ComS 575X: Computational Perception
Homework 1, Part III (Basic OpenCV)
Out: Wed. Jan 16, 2008
Due: Mon. Jan 30, 2008
Submit this part in electronic form (see details below).

To introduce you to OpenCV, we will do three fairly simple tasks. For this part you must submit your results in electronic form on your section of the class Wiki. The TAs have prepared some templates that you can use to simplify your task (check the wiki page).

1. **Image Negative**

   a) Start with a color JPEG image of yourself. If you don't have one already find a webcam and take one. Now write a complete C/C++ code using openCV that reads your image, converts it to grayscale, inverts the grayscale image to produce the negative, and then saves the negative to another JPEG file. Post your original image, your edges image, and your code on your section of the wiki.

   b) Use tha same image as in a) to produce three other images which contain the Red, Green, and Blue components of the original image.

2. **Cropping and labeling a video**

   For this task you must first find a digital movie (on Windows you can only use the AVI format; on Linux you can use MPEG if you compiled openCV with ffmpeg; see the online tutorials for more details). If you don't have a movie we posted some that you can download from the wiki page. Once you have the movie, write a program that uses openCV and reads the video one frame at a time, resizes the image to 1/2 the original size (1/4 the original area), prints your name on each frame, and saves the resulting video to another file.

   Post your original video, your results video, and your code on your wiki page. If you choose to use your own video please choose a small one ($\leq$ 2M) to save web space. If all else fails, just compress your videos in zip or tar.gz files before you upload them.

3. **Combining Two Videos: Picture-In-Picture**

   This task is very similar to the previous one. This time, however, you must read from two video files at the same time and combine them into one movie. The background of the resulting movie should come from the first video; the picture-in-picture part from the second video. Also, add a slider to your openCV window which controls the width (in pixels) of the screen occupied by the picture-in-picture window. Your application should be able to save the movie while you actively change the width parameter by dragging the slider with the mouse. The initial value of the slider should be set to 50% of the screen width. The initial position of the picture-in-picture window should be in the upper-left corner.

   Post only the resulting movie and your code on the wiki page.

   That's it. Good Luck!