Computer Vision (600.461/600.661)
Homework 2: Color and Image Processing

Instructor: René Vidal

Due 09/23/2014, 11.59PM Eastern

1. **(20 points) Image filtering, enhancement, and edge detection**.

   (a) Study MATLAB functions `imread`, `brighten`, `contrast`, `histeq`, `imcontrast` and `imadjust`. Write a script `hw2q1a.m` that applies each one of these functions to the MATLAB image `peppers.png`. Plot the original and the transformed images and comment on what each function does.

   (b) Study the functions `imnoise`, `medfilt2`, `conv2`, `filter2`, `fspecial`, `imfilter`, and `edge`. Write a script `hw2q2b.m` that does the following. Load image `peppers.png`. Convert it from RGB to grayscale using the function `rgb2gray`. Add salt and pepper noise to the image. Filter the resulting image using a 3x3 mean filter, a 3x3 median filter, and Gaussian filter with $\sigma = 1.5$ pixels. Repeat, but this time add Gaussian noise with $\sigma = 1$ in the $[0, 255]$ range ($\sigma = 1/256$ in the $[0, 1]$ range) instead of salt and pepper noise. Plot each one of the images and comment on what works best.

   (c) Write a script `hw2q1c.m` that does the following. Load image `peppers.png`. Convert it from RGB to grayscale using the function `rgb2gray`. Find the edges in the image using the MATLAB function `edge`. Use the following methods: Sobel, Prewitt, Roberts, Laplacian of Gaussian and Canny. Compare your results.

2. **(30 points) Color-based face detection.** One way to detect faces in color images is to search for pixels that have a skin-like color. The figure below shows an example where clusters t_4, t_5 and t_6 of the normalized RGB color space represent primary and secondary face colors. In this exercise, you will implement this simple color-based faced detection algorithm.

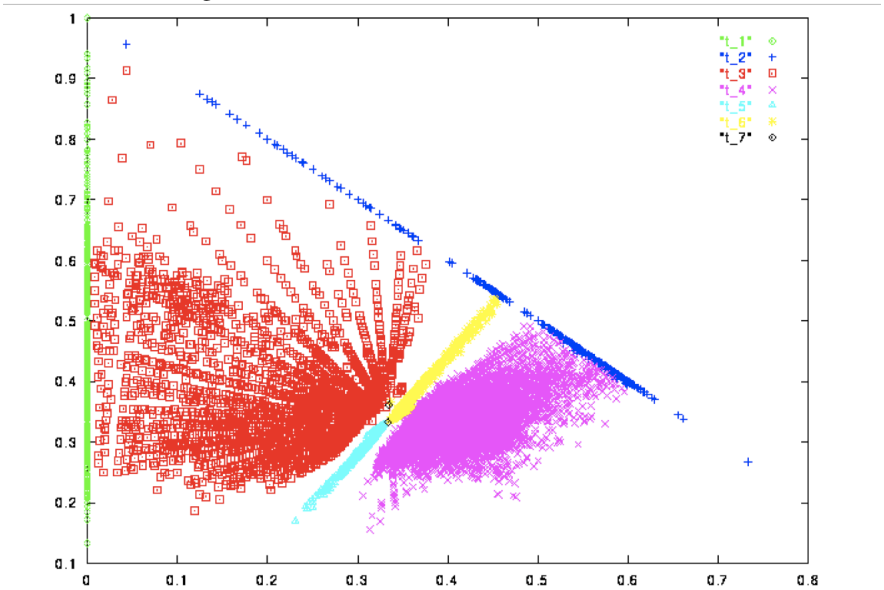

Figure 6.12: Skin color clusters obtained from training: the horizontal axis is R_{norm} and the vertical axis is G_{norm}. The cluster labeled as `t_4` is the primary face color, clusters `t_5` and `t_6` are secondary face clusters associated with shadowed or bearded areas of a face. (Figure from V. Bakic.)

   (a) Study the following MATLAB functions `colormap`, `hsv2rgb`, `rgb2gray`, `rgb2hsv`, `rgb2ntsc`, and `rgb2ycbcr`, and write a sentence or two about what each function does.

(b) Download the image http://www.allthetests.com/quiz19/picture/pic_1151313471_10.jpg. Convert the RGB coordinates to HSV coordinates and plot the following grayscale images: R, G, B, H, S, V. Comment on what you see, e.g. in which regions of which image skin color is more visible.

(c) Convert the RGB coordinates to normalized rgb coordinates. Plot g versus r as in Figure 6.12. Find a region of the r-g space that corresponds to the face colors. That is, find a function f (maybe linear as shown in Figure 6.12) such that face pixels can be determined by a rule of the form

$$\text{pixel } (x, y) \text{ is a face pixel if } f(r(x, y), g(x, y)) \geq 0. \tag{1}$$

You can do this by trial and error (we will learn automatic methods later in the course). For instance, choose a function f, define the mask

$$M(x, y) = \begin{cases} 1 & \text{if } f(r(x, y), g(x, y)) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and plot the image $J(x, y) = I(x, y)M(x, y)$. Repeat this until faces are "correctly" detected in $J(x, y)$.

(d) Repeat part 2c using the hue and saturation coordinates of the HSV color representation. If $H(x, y)$ and $S(x, y)$ are respectively the hue and saturation at pixel (x, y), then the rules to determine a face pixel may be of the form $a \leq H(x, y) \leq b$ and $c \leq S(x, y) \leq d$.

(e) Apply the rules you learned in parts 2c and 2d to the test image http://i168.photobucket.com/albums/u196/thesixfriends/friends45698.jpg. Which color representation works better and why?

Submission instructions. Send email to vision14jhu@gmail.com with subject 600.461/600.661:HW2 and attachment `firstname-lastname-hw2-vision14.zip` or `firstname-lastname-hw2-vision14.tar.gz`. The attachment should have the following content:

1. A file called `hw2.pdf` containing your answers to each one of the analytical questions. If at all possible, you should generate this file using the latex template hw1-vision14.tex. If not possible, you may use another editor, or scan your handwritten solutions. But note that you must submit a single PDF file with all your answers.

2. For coding questions, submit a file called `README`, which contains instructions on how to run your code. Use separate directories for each coding problem. Each directory should contain all the functions and scripts you are asked to write in separate files. For example, for HW2 the structure of what you should submit could look like

   (a) `README`

   (b) `hw2.pdf`

   (c) `hw2q3`: `hw2q3c.m`, `hw2q3e.m`

   (d) `hw2q4`: `hw2q4b.m`, `hw2q4c.m`

   The TA will run your scripts to generate the results. Thus, your script should include all needed plotting commands so that figures pop up automatically. Please make sure that the figure numbers match those you describe in `hw2.pdf`. You do not need to submit input or output images. The output images should be automatically generated by your scripts so that the TA can see the results by just running the scripts. In writing your code, you should assume that the TA will place the input images in the directory that is relevant to the question solved by your script. Also, make sure to comment your code properly.