ECE 172A MATLAB Homework #3A
Due Friday 10/24/08
Red-eye detection and removal – Mini Project

Reference : B. Smolka, K. Czubin, J. Y. Hardeberg, K. N. Plataniotis, M. Szczepanski, K. Wojciechowski, "Towards automatic redeye effect removal", Pattern Recognition Letters Volume 24, Issue 11, July 2003, Pages 1767-1785.

In this Matlab assignment, we will implement some parts of the red eye removal algorithm presented in the above paper. We will test the implementation on a small set of images.

1. Database

For this project we will work with a small set of 5 images downloaded from the internet. These .jpg images are available on the class website. Download them to your Matlab working folder. Note that the images are all of different sizes. As such, all your Matlab code should be capable of reading and processing images of arbitrary sizes.

2. Search-space reduction – Skin tones

The first step in detecting the red-eye in an image is to use some heuristics to reduce our search space, that is, instead of searching the entire image, we search a subset of the pixels for possible candidate red-eyes. One way of doing this is to restrict the search to areas that have the color of the human skin.

a) Implement the RGB to HSV color space conversion described in section 2.1.2 of the paper. The output should be an image with the HSV values stored in the 3 channels instead of the RGB values.
b) Use the thresholds given by eqn (7) in the paper to segment the skin/non-skin regions. The output will be a binary image with zeros representing non-skin pixels and ones representing skin pixels.
c) Clean the detected skin regions by using morphological operations explained in the class, especially the opening and closing operations to delete very small skin areas and fill in gaps in larger skin areas.

Required output: For test1.jpg, display the detected and cleaned skin regions as a black and white image.

3. Red-eye enhancement

Implement the RGB to YCbCr conversion technique described in section 2.3, eqns. (8) and (9) of the paper. Display the Cr component as a gray-scale image using the appropriate color map. The red pixels in the original image should appear brighter than the non-red images in this gray-scale image.

Required output: For test1.jpg, display the Cr gray-scale image obtained by applying (8) and (9) to the RGB image.

4. Pupil detection

Use 2D convolution with the circular edge finding filters described in figure 9 to detect the pupil regions in the gray scale image from the previous step. This filtering process will yield high values at those pixels that correspond to the center of circular disks. Hopefully the pupil of the red-eye is included in such pixels. The output will be another gray scale image. By using suitable thresholds on this new image, we can obtain some candidate centers for the pupil of the eyes. Use the skin regions to eliminate as many false candidates as possible. Experiment with different thresholds.

(Note : For simplicity, we will not use masks of varying sizes but fix the radius of the mask (outer circle) to 11 pixels)

The candidate pupil regions will be the union of the disks of radius 11, centered at the candidate centers, obtained above. This can be implemented as a convolution… think!

Required output: For test1.jpg, display the binary image corresponding to candidate pupil regions.

5. Red-eye replacement

The last step is to replace the redeye region of the pupil with the corrected pixel values. There are many ways of doing this and again, to keep things simple, we replace the Red channel of the pixels with the average of the Green and Blue channels. $R(I,j) = B(I,j)+G(I,j)$ for i,j in the candidate pupil regions.

Note : Depending on your threshold, you might have a lot of false candidates (false-positives). Replacing these pixels with the "corrected" values will lead to unwanted artifacts in the image. This can be minimized by eliminating the skin colored regions from the pupil candidate regions. For this purpose use the original skin pixels, not the ones obtained after morphological cleaning.

Required output: For all the test images, display the corrected images in color, both before and after using the skin pixel information for reducing the false positives.

This assignment is designed to give you a lot of experience in designing and implementing practical algorithms and hence be a good introduction to your actual 172A project. However guidance will be provided during the discussion section and office hours to clarify any doubts that might arise. Good luck!