EE435: Biometric Signal Processing

Project 6: Video Processing--Pupil Segmentation

Assigned: Fri 2/25/11 Due: Fri 3/04/11

This project builds on project 5, which you completed by creating a color iris image out of a grayscale iris image, where the boundary of the pupil was denoted as a red circle; this is indication of pupil segmentation. In this project the first part has you do a very similar thing (pupil segmentation), but it will be done on a video file, and the segmentation must be denoted in each image frame.

Pupil Segmentation in Video

Obtain the ee435_proj6_iris.avi video file from the professor. Write a MATLAB program that will take this video, analyze it, and write out a new video file that clearly shows the segmented pupil that is present in each frame using the steps that follow. The new video file will consist of the old video with the detected pupil indicated with a red circle drawn around it. The video that is written out should run at the same speed (frames per second) and have the same rows/cols dimensions as the original AVI file, but is now color.

Some potentially useful functions (not a comprehensive list):
aviinfo—provides pertinent information about the contents of an AVI file
aviread—allows you to read in an AVI file into an AVI object
movie2avi—allows you to write out an AVI file from a movie object
regionprops—determines the properties of a binary object, such as bounding box, centroid, etc.
avifile—allows you to create an AVI object

1. Use the `aviinfo` function to find the following information about this video file:

   Number of frames in the video: _____

   Dimensions (rows x cols x depth): _____

   Frames per second: _____

   At this frame rate, how long does it take to play: _____

   Any video compression ? (yes/no): _____

   Megabytes of storage required: _____

2. Read in the image file into a MATLAB avi object using the `aviread` function. When you run the command like the one below on a hypothetical avi file:

```
>> a=aviread(('ee435_proj6_hypothetical.avi'))
a =
        1x170 struct array with fields:
         cdata
        colormap
```

   In the avi video structure, each frame is stored with a *cdata* component and a *colormap* component. The *cdata* component is the actual color image frame data, and the *colormap* component contains the colormap for each frame. A colormap is a lookup table that cross references numeric values of pixels to what the red, green and blue component values are. By default, anything MATLAB displays uses 256 colors, so the colormap is a 256 x 3

matrix. The matrix determines the amount of red, green and blue to combine in each of the 256 colors that are displayed. The colormap is NULL ([ ]) for true color 24-bit images (otherwise, the colormap matrix would have 2^{24} –which is millions—of rows).

To get to any of the individual image frames and view it after reading in the avi file, for example the 37th frame, you would type commands like:

```
>> fr = a(37).cdata;, figure(1), imshow(fr)
```

When this is done, *fr* is now a variable in your workspace that is treated like an image (it could be color or grayscale). You can perform image processing with it.

3. Make a copy of the avi object…your processing will happen on this video vice the original video. For example, if you read in the avi object as variable a, use a command like `a1=a;` .Now write code to now process each frame of *a1* and display results for each frame. You should create a `for` loop to run through all frames. Your loop should perform the following steps:

   - Grab the frame from *a1* using a command like the one above that created variable *fr*.
   - Convert the color frame to grayscale using `rgb2gray`. It is less intensive to work on a grayscale image than a color image.
   - Segment the pupil like you did on Project 5 using this grayscale image.
   - Once you've determined the center and radius of the pupil, create a logical binary shifted circle array (using `create_shifted_circle` from Project 5) and insert a red circle in that frame of *a1*.
   - Before your loop ends, insert a 2x2 subplot that displays (1) the original frame; (2) the double image that has been inverted and squared, (3) the binary image that contains one object…the pupil; and (4) the color pupil image that has the red circle drawn in. When you run your program, have your professor initial that you showed it to him.

   Professor initials: _____

4. Now take your result and write out your own video file. This can be done with the following steps:

   - In your program's loop, comment out the line(s) that created your 2x2 subplot.
   - Instead of a subplot, `imshow` your frame that has the red circle drawn in it. Immediately after that line, insert a line as follows. In this line, *k* is the for loop looping variable.

     ```
     >> M(k) = getframe;
     ```

     This line of code will create an avi movie object that includes this frame (that is, it grabs the cdata and colormap as if it were a movie frame).

   - After your loop ends, insert a line of code that will write out this avi object to a .avi file. Use the `movie2avi` function, with no compression, and using the same frame rate as the original .avi file. Call your movie xxxx.avi, where xxxx is your last name(s).

   - Demonstrate your success by playing the movie for your professor

   Professor initials: _____

For this project: Turn in this sheet (filled out) along with your program that read in and processed the avi file.