STONY BROOK UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE 527: INTRO. TO COMPUTER VISION

Handed out: February 3, 2009 **Due on:** February 10, 2009

Problem Set #1

Submit the homework by email to cse527@cs.sunysb.edu

Problem 1: Horn, Exercise 2-6

What is the focal length of a compound lens obtained by placing two thin lenses of focal length f_1 and f_2 against one another? Hint: Explain why an object at distance f_1 on one side of the compound lens will be focused at a distance f_2 on the other side.

Problem 2: Color Statistics - *Implement and experiment with a Skin-Color Tracker.*

This could be one possible first step in a vision based body tracking system. The input is a video recording. The output is a sequence of binary images of the same dimension. Your program will set the pixels of the output image sequence to 1 for skin regions, and set to 0 everywhere else. Here are example matlab functions you can use. If you don't fully understand the matlab functions, use the matlab help feature. (for instance: **`help aviread, help aviinfo`**).

Here is the "recipe" :

- In matlab you can load a sequence of frames with the **`aviread`** function. For example load the first 20 frames with:

  ```
  mov = aviread('subject01.avi',[1:20]); %loads the first 20 frames
  ```

  (If your source is a quicktime movie, just convert it to an AVI)

- You can "digout" the first frame from the **`mov`** structure with: **`im = mov(1).cdata;`**
- You can display that image with: **`image(im); axis image;`**
- Now select a "training set" of skin pixels. You can do that with: **`skin_mask = roipoly;`** (and select with the mouse a skin region)
- To get a list of pixel indices of the mask area use: **`skin_inds = find(skin_mask>0);`**
- For better performance mark several training sets across different subjects.
- Use all those labeled skin pixels to estimate a multivariate Gaussian (as discussed in class and the first online paper describes). First build a matrix that contains the RGB values of the labeled skin pixels:

  ```
  skin_R = double(im(:,:,1));
  skin_G = double(im(:,:,2));
  skin_B = double(im(:,:,3));

  % skin data
  skin_data_rgb = [skin_R(skin_inds),skin_G(skin_inds),skin_B(skin_inds)];
  % the entire image data
  all_data_rgb = [skin_R(:),skin_G(:),skin_B(:)];
  ```

  To calculate the mean and the covariance of the skin Gaussian, use the matlab function:
  `skin_MN = mean(skin_data_rgb); skin_CV = cov(skin_data_rgb);`
- Now you are going to compute for the entire image (and successive images in the video) the probabilities that a pixel contains a skin region.
- Write a matlab function **`P = gaussdensity(all_data_rgb,MN,CV);`**
  If the input (all_data_rgb) is a Nx3 matrix, the output is a Nx1 matrix of probability values.

Use following formula for the gaussian:

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-1/2(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

If you do it in a smart way in matlab, you can fully vectorize it (no for-loops, just matrix inputs and matrix outputs).

- You can reshape the result into a 2D layer with:

```
[rows,cols,d] = size(im);  L1 = reshape(P,rows,cols);.
```

You can do the same for a "background layer" `L2` (using a different gaussian trained on background pixles). The last layer should be an "outlier layer" that has constant probability for each pixel. For example,

```
L3 = (ones(rows,cols)/256).^3;.
```

The final output of your algorithm should be normalized layers (posteriories). You get them in normalizing the values of each layer such that they add up to **1** for each pixel location (L1(x,y)+L2(x,y)+L3(x,y)=1). Do that with

```
S = L1+L2+L3; L1 = L1./S; L2 = L2./S; L3 = L3./S;
```

(Assuming uniform prior probabilities, those values could be interpreted as "posteriori probabilities": P(skin | RGB). And the un-normalized L1 values as "conditional density": p(RGB | skin).)

- You can visualize your results in generating a video of the skin layer (L1). Use the matlab function: **`avifile`**
- `Write out the resulting movie and display it on your web page.`

Scoring:

Full credit – implement and turn in a skin detection algorithm. Create a web page with your resulting movies. Provide a link to your webpage.
Students in the grad version of the class are required to do the extra credit.