EE 569: Homework #2

Issued: 9/12/2014 Due: 11:59PM, 10/12/2014

Problem 1: Edge Detection (40%)

In this problem, you are required to implement two basic edge detectors taught in class. Then, you will implement another edge detector, XDoG, to create an artistic visual effect.

(a) Basic Edge Detectors (Basic: 16%)

Implement two edge detectors and apply them to clean and noisy Elaine images as shown in Figure 1 (a) and (b).

1) Sobel Detector
   i. Normalize the x-gradient and the y-gradient values to 0-255 and show the results.
   ii. Tune the thresholds (in terms of percentage) to obtain your best edge map. An edge map is a binary image whose pixel values are either 0 (edge) or 255 (background).
2) LoG Detector
   i. Normalize the LoG response to 0-255 and show the results.
   ii. The LoG histogram is in form of a sharp symmetric spike. Develop an algorithm to determine the two knee locations of the LoG histogram. Then, you can use them as the threshold values to partition the LoG histogram into three values -1, 0, and 1. In order to show the corresponding ternary map, you can map -1, 0, 1 to gray-levels 64, 128, 192.
   iii. Apply zero crossing to obtain your best edge map.

Compare the results obtained in (1) and (2).

(a) (b)

Figure 1: (a) Original Elaine image and (b) noisy Elaine image.

(b) Creating Artistic Portrait Image (Advanced: 24%)

Implement part of the algorithm in [1] to make an artistic portrait image with the following steps:

1) *Artistic Portrait Background Generation via Bilateral Filtering*
   Apply *n* times bilateral filtering to different color channels of the portrait image under the *RGB* and the *CIELab* color spaces (*n* is often set to 3 or 4). It is claimed in [1] that it is preferable to apply the bilateral filter in a perceptually uniform feature space such as *CIELab*. An example of Lena image after the CIELab bilateral filtering process is shown in Fig. 2. Show and compare the filtering results of the test "Scarlett" image using the *RGB* and the *CIELab* color spaces. Do you agree with the claim in [1]?

Note: For the Matlab user, you can use the following statements to convert RGB to CIELab

```
colorTransform = makecform('srgb2lab');
lab = applycform(rgbImage, colorTransform);
```

and the following statements to convert CIELab to RGB.

```
colorTransform = makecform('lab2srgb');
rgb = applycform(labImage, colorTransform);
```

For the C user, you can use the package from the following website to conduct conversion between RGB and CIELab space.
http://www.getreuer.info/home/colorspace#TOC-Compiling

(a) (b)

Figure 2: (a) Lena image and (b) its filtered image result.

2) *Artistic Portrait Contour Generation via XDoG*

To generate the artistic contour of a portrait image, we will use the XDoG operation [1]. Recall that the traditional DoG operator is defined as:

$$g(\hat{x},\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\|\hat{x}\|^2}{2\sigma^2}} \tag{1}$$

$$G(\hat{x},\sigma,I) = \frac{1}{2\pi\sigma^2} \int I(x)\, e^{-\frac{\|\hat{x}-x\|^2}{2\sigma^2}}\, dx \tag{2}$$

$$D_0(\hat{x},\sigma,k,I) = G(\hat{x},\sigma,I) - G(\hat{x},k\cdot\sigma,I) \tag{3}$$

where $g(\hat{x},\sigma)$ is the Gaussian kernel centered at location $\hat{x}$ with variance σ, $G(\hat{x},\sigma,I)$ is the convolution of image I and $g(\hat{x},\sigma)$, D_0 is the DoG response, and where k is a factor relating the radii of the standard deviations of the two Gaussian functions. Usually, k is set to 1.6 so that the shape of DoG is similar to that of the LoG operator. The XDoG operator is defined as:

$$D_X(\sigma, k, \tau) = G(\sigma) - \tau \cdot G(k \cdot \sigma) \tag{4}$$

$$E_X(\sigma, k, \tau, \epsilon, \varphi) = \begin{cases} 1, & \text{if } D_X(\sigma, k, \tau) < \epsilon \\ 1 + \tanh(\varphi \cdot (D_X(\sigma, k, \tau)), & \\ \quad \text{otherwise.} & \end{cases} \tag{5}$$

where $\hat{x}$ (co-ordinate) and I (source image) are omitted for simplicity. Additional parameters in (4) and (5) have the following functions:

- Parameter ε shifts the detection threshold, thereby controlling sensitivity.
- Parameter τ changes the relative weighting between the larger and smaller Gaussians.
- The *tanh* function creates an adjustable soft ramp between the edge and non-edge values, with parameter φ controlling the *steepness* of this transition".

Play with these parameters to achieve the desired effect. An exemplary Lena contour image is shown in Figure 3. Apply your XDoG filter to the test "Scarlett" image and show the final result. Please specify the parameter values used in your design.

Figure 3: Lena's contour image

3) *Integration of Portrait Background and Contour*

As the final step, you should integrate the filtered background image and the contour image obtained in the previous two steps into one single output. This is achieved by first normalizing your contour image to the range of [0, 1]. Then, for each color channel of filtered image, conduct the following pixel-wise multiplication:

$$O(x, y, i) = B(x, y, i) \times C(x, y)$$

where (x, y) is the pixel location, i is the channel index, O, B, C denote the output image, the filtered background image and the contour image, respectively. The resulting artistic Lena image is shown in Figure 4.

Figure 4: Artistic Lena image

Conduct the integration operation on the test Scarlett image as shown in Figure 5. (Note: It is a plagiarism if you use any codes from the Internet.)

Figure 5: Scarlett.raw

4) *A short Cut?*
Tommy Trojan argued that, instead of using the XDoG operator, one can simply apply the DoG operator to obtain the edge map in the 2nd step and integrate the filtered background image and the edge map to create the artistic effect (*i.e.,* keeping steps 1 and 3 the same as above). Do you agree with him? Why? Test his idea on the test Scarlett image and discuss the differences between the two approaches.

Problem 2: Morphological Processing (30%)

Human fingerprints are unique and difficult to forge, and this characteristic allows authorities to identify human individuals in a crime scene or at the border entrance. The structural characteristic of a fingerprint is a pattern of ridges and valleys. In fingerprint images, ridges are dark whereas valleys are bright. Answer the following questions for the two fingerprint images in **Figure 6** using a combination of various morphological filters. Note, for bad fingerprint image, several ridges are weak connected because this finger is a little bit dry.

(a) fingerprint_good.raw

(b) fingerprint_bad.raw

Figure 6: Two fingerprint images: (a) a good one and (b) a bad one.

1) Fingerprint Image Binarization and Enhancement
   i. Develop a pre-processing procedure to enhance the two fingerprint images in Figure 6. Describe your algorithm and show the processed images.
   ii. Design a binarization algorithm that converts the two gray-level images to binary images. Describe and justify your design and show the binary (black/white) fingerprint images.
   iii. Develop a post-processing algorithm and apply it to the binary images to improve their quality as much as possible.

2) Fingerprint Characterization
   i. Find the total number of ridges, including long ridges, short ridges and isolated dots, for the two binary images obtained at the end of Step (1). (Hint: Use the pattern tables in the supplemental PDF file, called patterntables.pdf, to realize morphological filters.)
   ii. Plot the histogram of the ridge length. Use it to separate ridges into three types: long ridges, short ridges, and isolated dots. Discuss how to select the threshold between them.
   iii. Remove isolated dots, connect broken ridges, and show the new fingerprint images.

3) Minutiae Extraction
   Minutiae play an important role in the fingerprint matching. Two types of minutiae (*i.e.*, the ridge bifurcation and the ridge termination) are shown in Figure 7**.** There should be **NO minutiae detected** in the fingerprint boundary.

   Describe an algorithm to extract the two types of minutiae for these two fingerprint images, and show their locations in the original images circled by different colors where the blue color indicates the ridge bifurcation and the red color indicates the ridge termination.

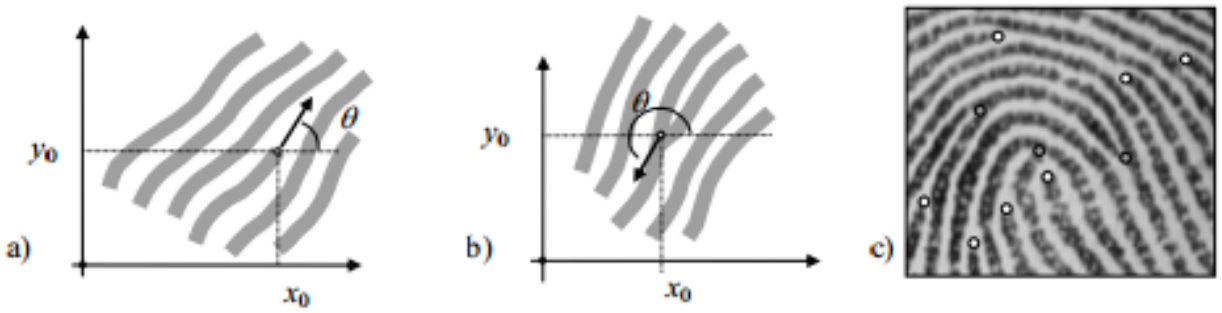

Figure 7: a) A termination minutia: $[x_0, y_0]$ are the minutia coordinates; θ is the angle that the minutia tangent forms with the horizontal axis; b) A bifurcation minutia: θ is now defined by means of the termination minutia corresponding to the original bifurcation that exists in the negative image; c) Termination (white) and bifurcation (gray) minutiae in a sample fingerprint

Problem 3: Digital Half-toning (30+10%)

(a) Dithering (Basic: 30%)

Implement the following four methods to convert boat.raw to half-toned images. There are 256 gray levels for pixels in boat.raw. In the following discussion, *F(i,j)* and *G(i,j)* denote the pixel of the input and the output images at position (*i,j*), respectively. Compare the results obtained by these algorithms in your report.

Figure 8: boat.raw

1) Fixed thresholding

Choose one value, T, as the threshold to divide the 256 levels into two ranges. An intuitive choice of T would be 127. For each pixel, map it to 0 if it is smaller than T, otherwise, map it to 255, i.e.

$$G(i,j) = \begin{cases} 0 & \text{if } 0 \le F(i,j) < T \\ 255 & \text{if } T \le F(i,j) < 256 \end{cases}$$

2) Random thresholding

In order to break the monotones in the result from fixed thresholding, we may use a 'random' threshold. The algorithm can be described as:

- For each pixel, generate a random number in the range 0 ~ 255, so called *rand(i,j)*
- Compare the pixel value with *rand(i,j)*. If it is greater, then map it to 255; otherwise, map it to 0, i.e.

$$G(i,j) = \begin{cases} 0 & \text{if } rand(i,j) \le F(i,j) \\ 255 & \text{if } rand(i,j) > F(i,j) \end{cases}$$

The built-in rand function in C/C++/Matlab generate numbers in uniform distribution.

3) Dithering Matrix

Dithering parameters are specified by an index matrix. The values in an index matrix indicate how likely a dot will be turned on. For example, an index matrix is given by

$$I_2(i,j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

where 0 indicates the pixel most likely to be turned on, and 3 is the least likely one. This index matrix is a special case of a family of dithering matrices first introduced by Bayer [4]. The Bayer index matrices are defined recursively using the formula:

$$I_{2n}(i,j) = \begin{bmatrix} 4 * I_n(x,y) + 1 & 4 * I_n(x,y) + 2 \\ 4 * I_n(x,y) + 3 & 4 * I_n(x,y) \end{bmatrix}$$

The index matrix can then be transformed into a threshold matrix T for an input gray-level image with normalized pixel values (*i.e.* with its dynamic range between 0 and 1) by the following formula:

$$T(x,y) = \frac{I(x,y) + 0.5}{N^2}$$

where N^2 denotes the number of pixels in the matrix. Since the image is usually much larger than the threshold matrix, the matrix is repeated periodically across the full image. This is done by using the following formula:

$$G(i,j) = \begin{cases} 1 & \text{if } F(i,j) > T(i \bmod N, j \bmod N) \\ 0 & otherwise \end{cases}$$

where *G(i, j)* is the normalized output binary image. Please create 2 × 2, 4 × 4 thresholding matrices and apply them to halftone the boat.raw image.

(b) Error Diffusion (Basic: 10%)

Implement the error diffusion technique with serpentine scanning, show the corresponding output image and discuss your result. Also, suggest ideas to get better results with justification (no implementation needed in this part.)

The Floyd-Steinberg (FS) error diffusion matrix [5] is in form of:

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

(c) Color Halftoning with Error Diffusion (Bonus 10%)

Figure 9: trees.raw

1) Separable Error Diffusion

One simple idea to achieve color halftoning is to separate an image into RGB three channels and apply the Floyd-Steinberg error diffusion algorithm to quantize each channel separately. Then, you will have one of the following 8 colors, which correspond to the 8 vertices of the RGB cube at each pixel:

W=(0,0,0), Y=(0,0,1), C=(0,1,0), M=(1,0,0),
G=(0,1,1), R=(1,0,1), B=(1,1,0), K=(1,1,1)

Note that (W,K), (Y,B), (C,R), (M,G) are complementary color pairs. Please show and discuss the result of the half-toned color tree image. What is the main shortcoming of this approach?

2) MBVQ-based Error diffusion

Shakad et al. [9] proposed a new error diffusion method to overcome the shortcoming of the separable error diffusion method. They partition the RGB color space into six Minimum Brightness Variation Quadrants (MBVQ) as shown in Fig. 2 of [9]. Then, the MBVQ-based error diffusion can be conducted as follows. Given the RGB value and the quantization error at a pixel located in (x, y). They are denoted by RGB(x,y) and e(x,y), respectively.

- Determine its MBVQ quadrant based on RGB (x, y);
- Find the vertex V of the MBVQ tetrahedron to closest RGB (x, y) + e (x, y) and output the value of V at location (x,y);
- Compute the new quantization error using RGB (x, y) + e (x, y) - V
- Distribute the quantized error to the future pixel error buckets e using a standard error diffusion process (e.g. the FS error diffusion).

You may refer to [9] for more details.

Implement this algorithm and apply it to the tree image in Fig. 9. Compare the output with that obtained in (1). Discuss the difference between these two methods.

Appendix:

Problem 1: Edge Detection			
elaine.raw	256x256	8-bit	gray
noisy_elaine.raw	256x256	8-bit	gray
Lena256.raw	256x256	24-bit	color(RGB)
Lena_bilateral.raw	256x256	24-bit	color(RGB)
Lena_contour.raw	256x256	8-bit	gray
Lena_artistic.raw	256x256	24-bit	color(RGB)
Scarlett.raw	400x300	24-bit	color(RGB)
Problem 2: Morphological Processing			
fingerprint_good.raw	388x374	8-bit	gray
fingerprint_bad.raw	388x374	8-bit	gray
Problem 3: Digital Half-toning			
boat.raw	512x512	8-bit	gray
trees.raw	350x258	24-bit	color(RGB)

References

[1] Winnemöller, Holger, Sven C. Olsen, and Bruce Gooch. "Real-time video abstraction." *ACM Transactions On Graphics (TOG)* 25.3 (2006): 1221-1226.

[2] http://www.easyrgb.com/index.php?X=MATH&H=02#text2

[3] Winnemöller, Holger. "XDoG: advanced image stylization with extended difference-of-Gaussians." *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2011.

[4] B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," SPIE MILE- STONE SERIES MS, vol. 154, pp. 139–143, 1999.

[5] R. W. Floyd, "An adaptive algorithm for spatial gray-scale," in Proc. Soc. Inf. Disp., vol. 17, 1976, pp. 75–77.

[6] J. F. Jarvis, C. N. Judice, and W. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," Computer Graphics and Image Processing, vol. 5, no. 1, pp. 13–40, 1976.

[7] P. Stucki, MECCA-A Multiple-Error Correction Computation Algorithm for Bi-Level Image Hardcopy Reproduction. IBM Thomas J. Watson Research Center, 1991.

[8] Maltoni, Davide. "A tutorial on fingerprint recognition." Advanced Studies in Biometrics. Springer Berlin Heidelberg, 2005. 43-68.

[9] D.Shaked, N. Arad, A.Fitzhugh, I. Sobel, "Color Diffusion: Error-Diffusion for Color Halftones", HP Labs Technical Report, HPL-96-128R1, 1996.

[10] http://images.google.com/

[11] http://sipi.usc.edu/database/