Project 2

CSE 402 - Biometrics and Pattern Recognition
Instructor: Dr. Arun Ross
Due Date: November 11, 2022 (11:00pm ET)
Total Points: 100

Note:

1. **You are permitted to discuss the following questions with others in the class. However, you *must* write up your *own* solutions to these questions. Any indication to the contrary will be considered an act of academic dishonesty. Copying from *any source* constitutes academic dishonesty.**
2. **A neatly typed report is expected (alternately, you can neatly handwrite the report and then scan it). The report, in PDF format, must be uploaded in D2L by November 11, 11:00 pm. Late submissions will not be graded. In your submission, please include the names of individuals you discussed this assignment with and the list of external resources (e.g., websites, other books, articles, etc.) that you used to complete the assignment (if any).**
3. **When solving equations or reducing expressions you must explicitly show every step in your computation and/or include the code that was used to perform the computation. Missing steps or code will lead to a deduction of points.**
4. **Code developed as part of this assignment must be (a) included as an appendix to your report or inline with your solution (in the report), and also (b) archived in a single zip file and uploaded in D2L. Including the code without the outputs or including the outputs without the code will result in deduction of points.**
5. **Please submit the report (PDF) and the code (Zip file) as two separate files in D2L.**

---

1. [20 points] Write a program that inputs a 3×3 matrix consisting of orientation field values and determines if the matrix corresponds to a singular point or not. If it corresponds to a singular point, determine if it is a core (loop) or a delta point.

   Input the following matrices to your program and report the output.

   1.

10	15	-10
12	0	15
13	12	-5

   2.

45	90	-50
50	0	-45
5	0	-5

   3.

50	0	-50
75	0	-70
85	90	-85

4.

45	2	-50
90	0	90
-50	2	50

2. [25 points] The ridge pattern in a local area of a finger can be approximated by a cosine wave:

$$w(x,y) = A\cos\left[2\pi f_0(x\cos\theta + y\sin\theta)\right].$$

Here, $w(x,y)$ denotes the pixel intensity at location (x,y). Generate and **display** ridge patterns, each of size 600×600, at the following orientation (θ) values: 0°, 45°, 90°, 135°. You may set $A = 80$ and $f_0 = 0.01$.

Now repeat the exercise, with $A = 160$ and $f_0 = 0.01$; $A = 80$ and $f_0 = 1$; and $A = 80$ and $f_0 = 10$. **What are your observations?**

3. [25 points] Using the gradient estimation method discussed in class (based on edge filters), write a program to compute the *orientation field* of a fingerprint image. The orientation should be computed for each pixel location. (So the number of rows and columns in the orientation field matrix should be the same as that of the image). Use the Sobel Operator to compute the x and y *gradient* value at each pixel location. Use a window size of 9×9 when computing the *orientation* field value associated with a pixel location (so value of k is 4). Run your program on the set of 10 fingerprint images available here.

Use the drawOrientation.m program (which is in Matlab) to display the orientation field as an overlay on the original fingerprint image. Include these overlay images in your submission.

Note: [1] You may not be able to compute the gradient values and the orientation field values on the border pixels. You may set those values to 0 or some other constant number in the orientation field matrix. [2] It would be better to use the atan2 function rather than the atan function. [3] You can use any programming language to compute *the orientation field. You can then use Matlab to* display *the orientation field based on the drawOrientation.m program.*

4. [30 points] Recall that a minutiae set, M, is a set of 3-tupled values $M = \{(x_i, y_i, \theta_i)\}$, $i = 1, 2 \ldots N_M$, where (x_i, y_i) is the location of minutiae i, θ_i is its orientation, and N_M is the total number of minutiae in M. Implement the minutiae matching method discussed in class (RANSAC method) that compares two minutiae sets M_1 and M_2, and outputs the transformation parameters t_x, t_y and t_θ relating M_2 with M_1, along with the number of matching minutiae pairs (you can use a tolerance value of 10 when determining matching minutiae pairs).

You are given a set of 10 minutiae files pertaining to 5 different fingers (the orientation value is denoted in "degrees"). Run your program on all possible pairings of the minutiae files and present a table listing the results in the following format: First File Name (M_1), Second File Name (M_2), t_x, t_y, t_θ, Number of Matching Minutiae Pairs (s). There will be 10 genuine pairings and 80 impostor pairings. For example, the first line of the table will be:

```
user001_1.minpoints user001_2.minpoints -65 -6 0.052360 19
```