COMP 766: Assignment 1
Available: Tuesday, February 11th, 2014
Due Date: Thursday, February 27th, 2014

This assignment considers the problem of finger print analysis use curve inference, exploring the ideas we have been discussing in class. Finger prints are a special class of challenging patterns for naive edge detection algorithms to analyze because the features have low contrast locally and the appear at very fine scales. The source image from this assignment can be downloaded from the following url:
`www.bromba.com/knowhow/Fingerprintonfinger.jpg`
You are highly encouraged to use Matlab for your implementations, for ease of prototyping and to be able to generate visualizations with ease.

1 Part I: (30%) INITIAL TANGENT ESTIMATES

Convolve the finger print image with the template discussed in Section IV, using the following 16 orientations: $n \times \pi/16, n = 0, 1, 2, 3, 4, 5, 6, 7, ..., 15$. In other words, you will perform 16 separate convolutions. Treat the constants $A, B, C, \sigma_1, \sigma_2, \sigma_3$ as inputs and select appropriate values for them by trial and error. One way to do this is to visualize the template so that the central bright line is roughly the thickness of the curve like features and it is extended in the vertical direction for an appropriate length (see Fig. 4 in the paper). Ofcourse, not all features have exactly the same thickness, so for a fixed set of choices for the constants you will be detecting patterns of a particular width.

Now normalize the sixteen outputs to get your assignment vector p_i at each location i in the image. Visualize these initial estimates by drawing vectors in the direction λ for those assignments for which $p_i(\lambda)$ is greater than some threshold, e.g., 0.15. You might want to overlay these vectors on the original input image. What you should see is a large set of tangent estimates, where a subset correspond to the local orientation of finger print curves.

2 Part II: (40%) CO-CIRCULARITY SUPPORT

Now, using the notions of compatibility developed in Section V: A, B, C, D of the paper, estimate the support $s_i(\lambda)$ for orientation λ at location i. More specifically,

$$s_i(\lambda) = \max_{k=1,K} \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}^{kk'}(\lambda, \lambda') p_j(\lambda'),$$

where

$$r_{ij}^{kk'}(\lambda, \lambda') = c_{ij}(\lambda, \lambda')K_{ij}^{k}(\lambda, \lambda')C_{ij}^{kk'}(\lambda, \lambda').$$

This notion of compatibility includes the ideas of co-circularity, curvature classes and consistency of curvatures, but **NOT** lateral maxima selection, developed in Section V of the paper. Your implementation of this support function will involve choices for several parameters, as well as a choice for the number of curvature classes to use. As well, it is reasonable to build the support function $s_i(\lambda)$ using a large enough but not infinite circular neighborhood of each pixel i. You can visualize your results by drawing vectors in the direction λ for those assignments for which $s_i(\lambda)$ is greater than some threshold. You should now expect to see some of the initial tangent estimates being suppressed. Ofcourse, the use of curvature classes and consistency of curvatures in defining the compatibility function really only makes sense as the assignments are iteratively updated, but this will happen in Part III.

3 Part III: (30%) RELAXATION LABELING

Using the initial assignment given by Part I and the co-circularity support given by Part II, compute the average local support

$$A(p) = \sum_{i=1}^{b} s_i(\lambda)p_i(\lambda).$$

Now, apply a few iterations of relaxation labeling to converge to a consistent labeling. Use the algorithm developed in the relaxation labeling paper, along with the radial projection update rule described in Appendix A. Visualize your results in the same way as before. Surviving tangents should now correspond to "true" finger print curves. Also, pay attention to what happens near bifurfcations and crossings. In principle, you should be able to pick these up as well.

Notes Whereas you are free to describe the details of your implementation in a high level form, and to provide these implementations on a website if you like, you are not required to turn in any code. Furthermore, there are many refinements developed in the articles (such as Section VI: B, D) that you should not necessarily worry about. I am looking for evidence that you understood most of the technical aspects of the framework and were able to use it, and that your report presents your ideas clearly and creatively. To this end you should turn in a brief (no more than 10 page) report. In your report you should:

1. Summarize your implementations and present your results for each part. It might help to set up images of your results on a website and include a link.

2. Explain any simplifying assumptions you have made and their consequences.

3. Explain your choices for the various parameters.
4. Provide some critical discussion of your results as well as of the overall framework.