ENCM 509 - Laboratory #4

Kyle Derby MacInnis

October 14, 2015

Abstract

This laboratory focuses on feature extraction of fingerprint ridges to help locate minutiae.

Introduction

This lab looked at feature extraction as it applied to fingerprint analysis. This lab is part of a larger number of future labs that will use the extracted data.

Procedure

1. Load in BMP image and convert to grayscale (intensity) image.
2. Run Lab4Fingerprint1.m and look at demonstration of feature extraction.

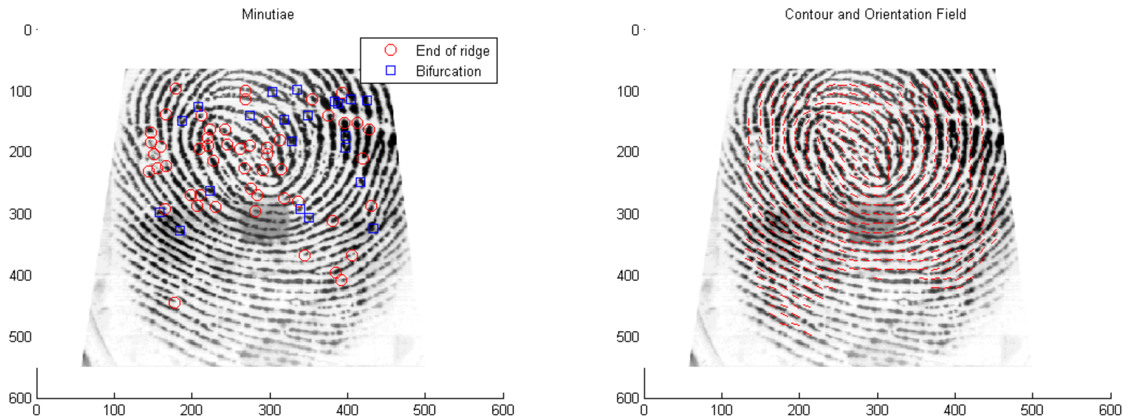


Figure 1: Feature Extraction used to find minutiae

3. Modify Lab4Fingerprint1.m and have it perform Histogram Equalization, and Denoising.

```
%% Exercise - Histogram Equalization
        figure(101);
        subplot(1,3,1);
        imhist(img);
        xlabel('Original Image Histogram');
        % Regular Histogram Eq.
        Himg = histeq(img);
        subplot(1,3,2);
        imhist(Himg);
        xlabel('Regular Histogram Equalization');
        % Adaptive Histogram Eq.
        aHimg = adapthisteq(img);
        subplot(1,3,3);
        imhist(aHimg);
        xlabel('Adaptive Histogram Equalization');
```

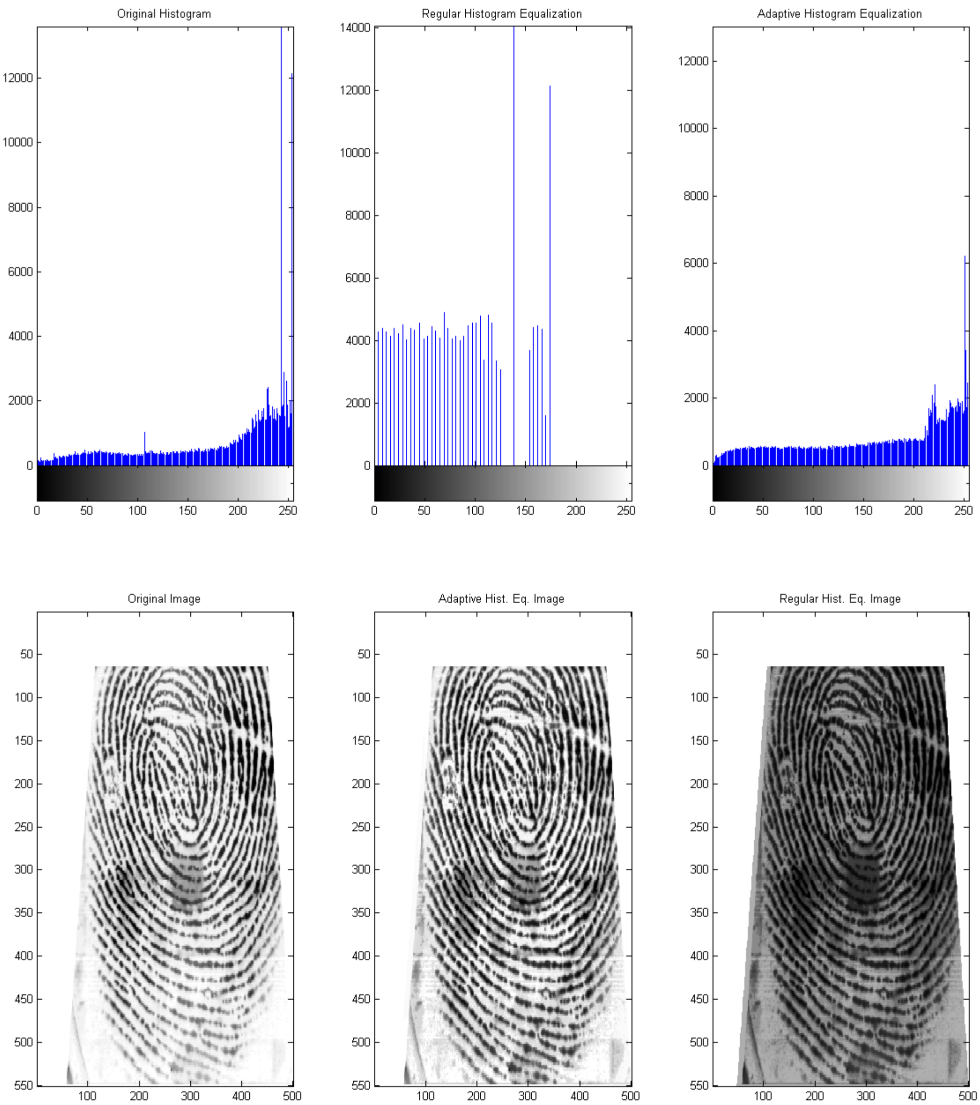
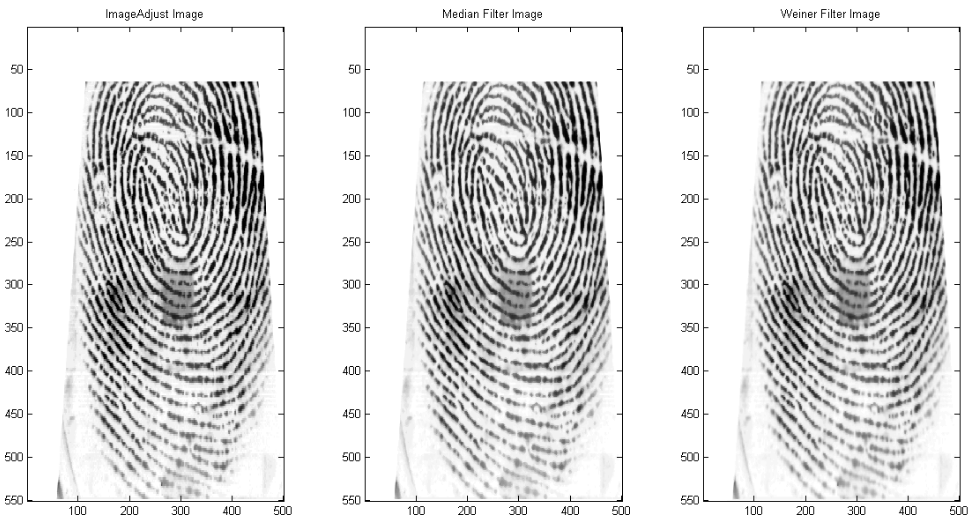


Figure 2: Histograms and Images for Histogram Equalization Enhancement

```
%% Exercise - Denoising Filters
        % Image Adjust
        Aimg = imadjust(img);
        Mimg = medfilt2(img);
        Wimg = wiener2(img);

        figure(102);
        subplot(1,3,1), imagesc(Aimg), colormap gray, title('ImageAdjust Image');
        subplot(1,3,2), imagesc(Mimg), colormap gray, title('Median Filter Image');
        subplot(1,3,3), imagesc(Wimg), colormap gray, title('Weiner Filter Image');
```



4. Inspect each part of the process and modify parameters to observe variation to the extracted minutiae and finished feature extractions.

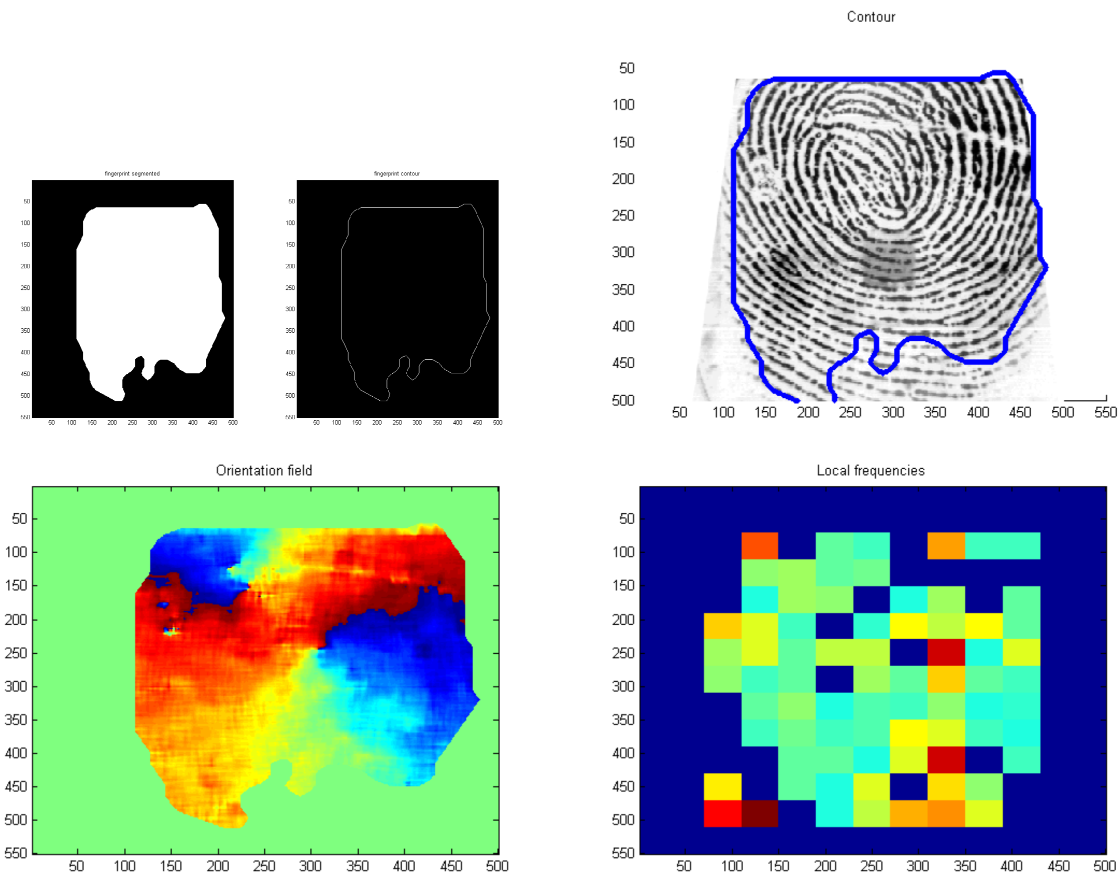


Figure 3: Various Steps used in feature extraction.

5. Perform Gabor Filtering and examine parameter changes.

```
%% Exercise Gabor Filtering
        % Gabor Filter Parameters
        x = 48; y = 32;
```

```
tx = 4; ty = 5;
angle = pi/4;
f = 0.10;
p = 36;
for i = 1:p
    angle = (pi/(p*(log(p)/log(i))))*(i);
    GabFlt = GaborFilter(x, y, tx, ty, angle, f);
    if i == 1
        Gabimg = (1/(p))*imfilter(img, GabFlt);
    elseif i > 1
        Gabimg = Gabimg + (1/p)*imfilter(img, GabFlt);
    end
    figure(104);
    imshow(Gabimg); colormap(gray);
end
```

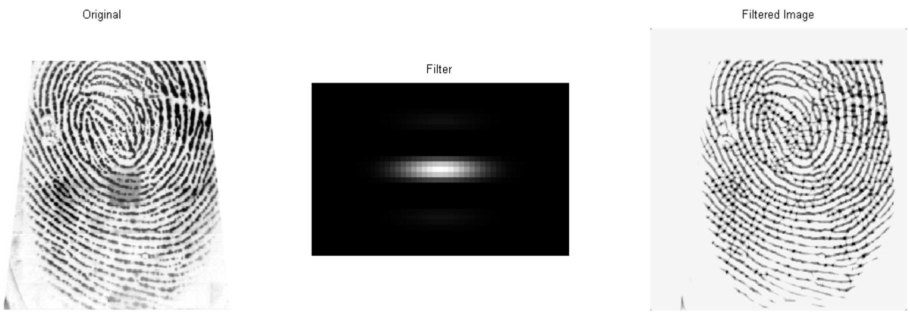


Figure 4: Gabor Filter and Filtered Image

Discussion

- Histogram Equalization appeared to slightly increase contrast using the *adaptive* method, but using the the regular method, contrast resolution did not improve and rather most of the image contrast was reduced.
- Denoising did not have much of an effect on any of the fingerprints I had collected. It had no noticeable effects.
- By varying the parameters of the segmentation function, the borders and the *smoothness* of the enclosed border was changed. Smaller values for results in a more jittery and sharp shape, whereas larger values produced more blob-like shapes and had smooth borders.
- There were three singularity points found in my finger print. However, there was five when I used the Gabor filtered image instead.
- Reducing the size of the parameters used in the Gabor filtering for the skeleton creation resulted in a generally *stronger* skeleton whereas larger values resulted in more convoluted skeletons.

Remarks on the Lab

This lab was quite easy to finish, and look forward to next lab.

Appendix A: MATLAB Code

Modified_Lab4Fingerprint1.m

```
%============ ENCM 509 ==============%
%============ Lab 4: Fingerprint: Part II ===============%
%============ Fingerpint matching score =================%
% by S. Yanushkevich, February 04,2009
% updated October 1, 2014

clear all
close all

disp('---------------------------------------------------------------------');
disp('                Fingerprint preprocessing and feature extraction');
disp('              Biometric Technologies Laboratory, UofC');
disp('---------------------------------------------------------------------');


%The image acquired using an optical fingerprint sensor is a gray-scale image saved in a
     bitmap or other image format file
%Traditionally, processing of a gray-scale fingerprint images include segmentation,
    ridge orientation estimation,
% a centre-of-mass or singularity point detection, ridge frequency estimation, Gabor
    filtering,
%thinning and skeleton cleaning, and, finally, minutiae detection

    %-----------------------------------------------------------------
    %-------- Reading image from a bitmap or other format
    %-----------------------------------------------------------------
disp('Reading the image from a file')
        [namefile,pathname]=uigetfile({'*.bmp;*.tif;*.tiff;*.jpg;*.jpeg;*.gif','IMAGE
    Files (*.bmp,*.tif,*.tiff,*.jpg,*.jpeg,*.gif)'});
        [img,map]=imread(strcat(pathname,namefile));
    img=rgb2gray(img);

%% ADD FILTERING SUCH AS
%% img1 = imadjust(img); img2 = medfilt2(img);
%%

%% Exercise - Histogram Equalization

        figure(101);
        subplot(1,3,1);
        imhist(img);
        xlabel('Original Image Histogram');
        % Regular Histogram Eq.
        Himg = histeq(img);
        subplot(1,3,2);
        imhist(Himg);
        xlabel('Regular Histogram Equalization');
        % Adaptive Histogram Eq.
        aHimg = adapthisteq(img);
        subplot(1,3,3);
        imhist(aHimg);
        xlabel('Adaptive Histogram Equalization');

%% Exercise - Denoising Filters
        % Image Adjust
        Aimg = imadjust(img);
        Mimg = medfilt2(img);
        Wimg = wiener2(img);

        figure(102);
        subplot(1,3,1), imagesc(Aimg), colormap gray, title('ImageAdjust Image');
        subplot(1,3,2), imagesc(Mimg), colormap gray, title('Median Filter Image');
        subplot(1,3,3), imagesc(Wimg), colormap gray, title('Weiner Filter Image');

%% Exercise Gabor Filtering
        % Gabor Filter Parameters
        x = 48; y = 32;
        tx = 4; ty = 5;
        angle = pi/4;
```

```
        f = 0.10;
        p = 36;
        for i = 1:p
            angle = (pi/(p*(log(p)/log(i))))*(i);
            GabFlt = GaborFilter(x, y, tx, ty, angle, f);
            if i == 1
                Gabimg = (1/(p))*imfilter(img, GabFlt);
            elseif i > 1
                Gabimg = Gabimg + (1/p)*imfilter(img, GabFlt);
            end
            figure(104);
            imshow(Gabimg); colormap(gray);
        end

        figure(103);
        subplot(1,3,1), imshow(img), colormap gray, title('Original');
        subplot(1,3,2), imshow(GabFlt), colormap gray, title('Filter');
        subplot(1,3,3), imshow(Gabimg), colormap gray, title('Filtered Image');

        Fp.imOrig = img;

         figure(1), subplot(1,3,1),imagesc(Fp.imOrig),colormap gray, title('Original
    Image');
         subplot(1,3,2), imagesc(aHimg),colormap gray, title('Adaptive Hist. Eq. Image')
    ;
         subplot(1,3,3), imagesc(Himg), colormap gray, title('Regular Hist. Eq. Image');
%% TO ADD FIGURE COMPARING FILTERS, USE
%% figure(1),
%% subplot 221, imshow(img); title('Original Image');
%% subplot 222, imshow(img1); title('imadjust');
%% subplot 223, imshow(img2); title('medfilt2');


    disp('(Press any key to continue)');
    pause;
    %------------------------------------------------------------------------
    %--------- Preprocessing: segmentation and countour
    %------------------------------------------------------------------------


            disp('Preprocessing: extracting fingerprint from background')
            Fp = segmentimage(Fp);

            figure(2),
            subplot(1,2,1), imagesc(Fp.imSegmented),colormap gray, title('fingerprint
    segmented');
            subplot(1,2,2), imagesc(Fp.imContour),colormap gray, title('fingerprint
    contour');

            [x,y]= find(Fp.imContour);
            figure(3), title('Contour'), hold on, imagesc(Fp.imOrig),colormap gray;
            plot(y,x,'.'), axis ij, axis([1 size(Fp.imOrig,1) 1 size(Fp.imOrig,2)]);
            hold off;

    disp('(Press any key to continue)');
    pause;
    % ------------------------------------------------------------------------
    %--------- Orientation field
    %------------------------------------------------------------------------

            disp('Compute orientation field');
            Fp = computeorientationarray(Fp);
            figure(4), imagesc(Fp.orientationArray), title('Orientation field');
            showorientationfield(Fp);

    disp('(Press any key to continue)');
    pause;
    %------------------------------------------------------------------------
    %--------- Finding the singularity point (central minutiae)
    %------------------------------------------------------------------------

            disp('Extracting the singularity point');
```

```
        Fp = findsingularitypoint(Fp);
        [x,y]= find(Fp.singularityArray);
        figure(5), hold on, imagesc(Fp.imOrig),colormap gray;
        plot(y,x,'.'), axis ij, title('Singularity Point');
        hold off;

disp('(Press any key to continue)');
pause;

%------------------------------------------------------------------------
%-------- Local ridge frequency
%------------------------------------------------------------------------

        disp('Compute local ridge frequency');
        Fp = computelocalfrequency(Fp, Fp.imOrig);
        figure(6), imagesc(Fp.frequencyArray), title('Local frequencies');


disp('(Press any key to continue)');
pause;

%------------------------------------------------------------------------
%-------- Gabor Filtering
%------------------------------------------------------------------------

        disp('Gabor filtering and enhancing skeleton structure');
        Fp = GaborEnhanced(Fp);
        figure(7), imagesc(Fp.imBinary), colormap gray, title('Binarized image');
%        figure(8), imagesc(Fp.imSkeleton), colormap gray, title('Skeleton');

         [x,y]= find(Fp.imSkeleton);
        figure(8), title('Skeleton');
        hold on, imagesc(Fp.imOrig), colormap gray;
        plot(y,x,'r.'), axis ij;
        hold off;

disp('(Press any key to continue)');
pause;




%------------------------------------------------------------------------
%-------- Thinning and skeleton cleaning
%------------------------------------------------------------------------


        disp('Morphological skeleton cleaning')
        Fp = cleanskeleton(Fp);
 %       figure(10), imagesc(Fp.imSkeleton), colormap gray, title('Skeleton');

        Sk=ones(size(Fp.imSkeleton,1),size(Fp.imSkeleton,2));
        Sk(1:size(Fp.imSkeleton,1)-50,1:size(Fp.imSkeleton,2)-50)=Fp.imSkeleton(26:
size(Fp.imSkeleton,1)-25,26:size(Fp.imSkeleton,2)-25);
        Sk = imcomplement(Sk);
        [x,y]= find(Sk);
  %      figure(11), title('Skeleton cleaning');
        figure(9), title('Skeleton cleaning');
        hold on, imagesc(Fp.imOrig),colormap gray;
        plot(y,x,'r.'), axis ij;
        hold off;

disp('(Press any key to continue)');
pause;
%------------------------------------------------------------------------
%-------- Finding minutiae
%------------------------------------------------------------------------

        disp('Extracting Minutiae')
        Fp = findminutia(Fp);

        [x1,y1]= find(Fp.minutiaArray==1);
        [x2,y2]= find(Fp.minutiaArray==2);
```

```
            figure(10),
%            hold on, imagesc(Fp.imSkeleton),colormap gray, title('Minutiae');
%            plot(y1,x1,'or','markersize',8);
%            plot(y2,x2,'sb','markersize',8), axis ij;
%            legend('End of ridge', 'Bifurcation');
 %           hold off;
%           figure(13),
            hold on, imagesc(Fp.imOrig),colormap gray, title('Minutiae')
            plot(y1,x1,'or','markersize',8);
            plot(y2,x2,'sb','markersize',8), axis ij;
            legend('End of ridge', 'Bifurcation');
            hold off;

     disp('(Minutiae array contains features to be used for fingerprint comparison in
    Lab 5)');
```