A Three Stage Ear Biometric System

Charles Chen (clchen@colorado.edu)
University of Colorado at Boulder

November 25, 2010

Table of Figures

1. Introduction:

The usage of ears as a biometric is a relatively new concept, compared to the more established method of facial recognition. Recent work has demonstrated that they can be used as a viable target for identification. Ears have several beneficial characteristics, such as a significant three dimensional structure and visual stability with age. Images of the ear can also be capture discreetly, as a target walks by, compared to facial recognition techniques that require a direct-on view of the target. Various methods have been posed to uniquely identify and match ear images to targets. Some borrow concepts from facial recognition, such as the "Eigen-Ear" methodology that derives from the "Eigen-Face" technique of facial recognition. Others utilize 3-dimensional surface data in order to determine ear structure. Although promising in their accuracy, depth based methods suffer from a scanning requirement. Of particular interest are methods that only require an image of an ear to extract, identify, and match features.

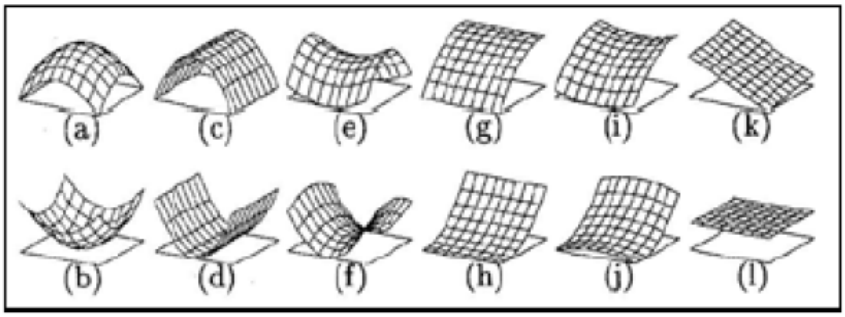
The ear as a biometric system proposed here will utilize image based techniques, as opposed to depth based techniques, to detect and identify ear images based on a set of references. It will also combine a wide variety of methods from the extensive body of relevant research on this topic.

2. Project Description:

The proposed ear biometric system can be divided into three main functions: detection and segmentation, feature processing, and matching.

2.1 Detection and Segmentation: Topographic Labeling

The first step of this biometric process is to detect and isolate images of the ear. Doing so reduces the amount of data that later stages must process. It also has the added benefit of removing background distractions that may interfere with the feature extraction and matching process. Instead of using a setup intensive trained detection method, such as Viola and Jones' cascaded Adaboost system, I have opted for a more lightweight and simpler approach. The Topographic Labeling method proposed by Milad Lankarany and Alireza Ahmadyfard provides this. Lankarany's method assigns a topographic label to each pixel based on the intensity gradient around it.



Topographic labels (a) peak (b) pit (c) ridge (d) ravine (e) ridge saddle (f) ravine saddle (g) convex hill (h) concave hill (i) convex saddle hill (j) concave saddle hill (k) slop hill and (l) flat

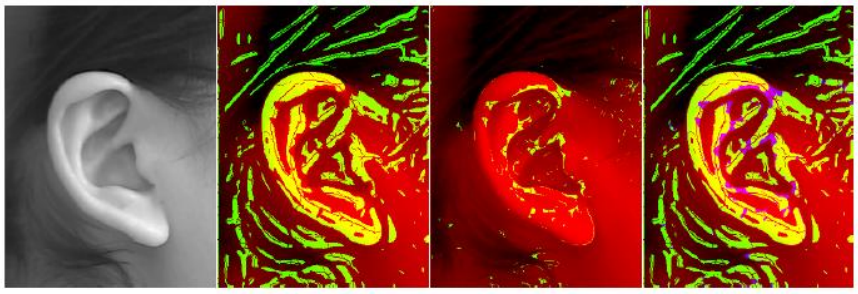
Figure 1: Topographic Label Types

Each group of labeled points provides a different set of highlighted features. Combining multiple label groups should allow for the selection of specific interest regions, such as the ear. This project utilized the same label choices as Lankarany, that is, the ridge, convex saddle hill, and convex hill labels.

Further thresholding removes stray labels and produces several clumped pixel regions.

Lankarany experimentally determined that the contour grouping that contained the greatest number of pixels is most likely to be the outer ear region. After looking at his test cases, this seems to be the case only when the image is already focused closely on the ear. Should the ear image be part of a larger and more varied background, his working assumption is not likely to be true. In some cases other distracting features contain the greatest number of pixels, such as patterned shirt collars or other textured regions. In other cases the portion of the mask attributed to the ear remains attached to a larger region around the side of the face, resulting in a poorly-isolated ear segmentation.

In order to make the detection more robust, this implementation further refines the label mask by multiplying it with the difference of the erosion and dilation of the image, then rethresholding the result.



(A). Selected image from USTB database, (B). Green part shows the convex hill label, (C). Green part indicates the convex sadlle hill, (D). Green part shows convex hill and blue part shows the convex saddle hill label, all extracted from selected ear image.

Figure 2: Assigned Topographic Labels

The determined ear segment is based upon the rectangular subregion around the group with the largest mean brightness value, operating under the assumption that the ear should have the most edges highlighted, and thus that subimage should have the highest mean brightness.

2.2 Feature Processing: Force Field Transform

The feature detection method is based heavily on the work of David Hurley, who proposed a feature extraction system based upon the Force-field energy transformation. This transformation is effectively a powerful smoothing filter. Using an analogy of mass and gravitation, each pixel is treated as a point mass, with its mass proportional to the brightness of that pixel. That pixel exerts a gravitational force upon every other pixel in the image. Summing the contributions of each pixel across the image creates a force-field, which can be further processed to find local minima, local maxima, or other features. Hurley suggests taking the divergence of the force-field and thresholding the resultant scalar field.

2.3 Comparison: Template Matching

The comparison stage of the process is a template-matching method, invariant to scale, translation, brightness, and contrast. This particular template-matching algorithem is based upon the work of Hae Yong Kim and Sidnei Alves de Araujo, whom proposed a novel method of rotation and scale invariant template matching. It is comprised of a three-tiered evaluation of candidate pixels, with each tier operating on the candidate pixels selected by the previous tier.

The first tier is a circular filter, which averages the grayscale values of all pixels at a radius r from some center pixel (x,y), for several values of r. The template image is re-sized to several scales s, and then sampled with the circular filter. The target image is sampled with the same set of radii, but at regular size. After both are sampled, a correlation is calculated at each pixel between its filtered value and each of the filtered values generated by the s scales. Of the s correlations calculated, the highest is chosen, with the scale corresponding to that correlation

becoming the probable scale of matching at that point. All points with a maximum correlation above a certain threshold are considered First-Grade candidate pixels, which are used for the second stage of testing.

First-Grade candidate pixels are then filtered using a radial sampling filter, which averages grayscale pixel values at long a ray of length r, at an angle of alpha away from the center pixel. The template and the target image are sampled for a list of angles alpha between -180 degrees and 180 degrees, then correlated. The maximum correlation is then found to determine the probable rotation angle of the template against the target pixel. Pixels with a maximum correlation above a second threshold are then upgraded to Second-Grade pixels.

After the application of the second filter, there should only be a small number of Second-Grade pixels. These sub-image around these pixels are then correlated against the scaled and rotated template (the scale and rotation being determined by the previous two steps). If this correlation is above a third threshold, t3, then the target pixel is considered to be a match to the template.

3. Procedure and Results

3.1 Detection and Segmentation

3.1.1 Initial Processing and Labeling

Before any calculations labels are assigned, the target image is first preprocessed. RGB images are converted to grayscale, and then blurred using a 5x5 Gaussian filter. After blurring the image is then contrast enhanced using the Matlab histeq() histogram equalization function.

Figure 3: Initial Image

Figure 4: After Preprocessing

Once preprocessed, the gradient and the Hessian matrix are calculated at every pixel of the image. Applying eigenvalue decomposition to the Hessian matrix produces the eigenvectors and eigenvalues of the Hessian matrix, which are used to determine the label that should be applied to the associated pixel.

$$\vec{\nabla} f = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right) \qquad \left\| \vec{\nabla} f \right\| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x} \right)^2 + \left(\frac{\partial f(x,y)}{\partial y} \right)^2}$$

$$H = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x y} \\ \frac{\partial^2 f(x,y)}{\partial y x} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix} = \begin{bmatrix} f^{(2,0)}(x,y) & f^{(1,1)}(x,y) \\ f^{(1,1)}(x,y) & f^{(0,2)}(x,y) \end{bmatrix}$$

$$H = UDU^T = [u_1 \;\; u_2].diag(\lambda_1 \;\; \lambda_2).[u_1 \;\; u_2]^T$$

Figure 5: Gradient and Hessian

Each pixel is then assigned a label corresponding to its topography.

Ridge:

$$\left\|\vec{\nabla} f(x,y)\right\| = 0 \quad , \quad \lambda_1 > 0, \lambda_2 = 0$$
$$\left\|\vec{\nabla} f(x,y)\right\| \neq 0 \quad , \quad \lambda_1 > 0, \vec{\nabla} f . \vec{u}_1 = 0$$
$$\left\|\vec{\nabla} f(x,y)\right\| = 0 \quad , \quad \lambda_2 > 0, \vec{\nabla} f . \vec{u}_2 = 0$$

Figure 6: Ridge Conditions

Convex Hill:

$$\left\|\vec{\nabla} f(x,y)\right\| \neq 0 \quad , \quad \lambda_1 > 0, \lambda_2 > 0$$
$$\left\|\vec{\nabla} f(x,y)\right\| \neq 0 \quad , \quad \lambda_1 > 0, \lambda_2 = 0$$

Figure 7: Convex Hill Conditions

Convex Saddle Hill:

$$\left\|\vec{\nabla} f(x,y)\right\| \neq 0 \quad , \quad \lambda_1 > 0, \lambda_2 < 0$$

Figure 8: Convex Saddle Hill Conditions

The three label sets for the target image are shown below.

Figure 9: Ridge Label

Figure 10: Convex Hill Label

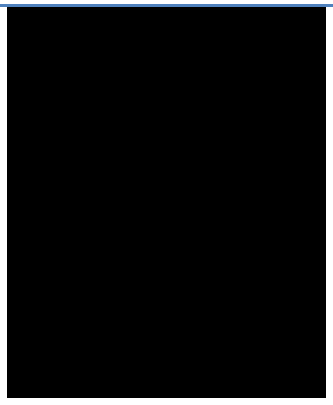


Figure 11: Convex Saddle Hill Label

These Images are then added together, and converted to a binary image such that non-zero pixels are assigned a value of one. In this case, the ear was not detached from the side of the face as necessary for isolation and segmentation. Another mask was calculated from the difference of the erosion and dilation of the target image. These two results were then multiplied together and rethresholded.

Figure 12: Result of Merged Labeled Pixels

Figure 13: Erosion and Dilation Difference

Figure 14: Combined and Thresholded Mask

3.1.2 Mask Refinement and Region Selection

In order to merge some of the smaller selected regions with the larger clusters, this mask is blurred and rethresholded ten times. Using the Matlab function bwlabel, we counted the number of distinct connected regions within the combined mask. If the number of connected regions is above a certain number (here a choice of 65), the merging process is performed repeatedly until this is not the case.

The bwlabel process is called again once the merging process is complete. For each connected region found, a rectangular subimage is cropped around the region and isolated for further processing.

Figure 15: After Region Merging

Figure 16: Colored Distinct Regions (Some Colors Repeated)

After ignoring excessively small regions (width or height < 40 pixels), the remaining subimage were processed using the difference of erosions and dilations, specifically using a triangular shape. The image was then filtered using the built in Matlab function stdfilt, which calculates the local standard deviation of each pixel in the image. The mean value of every pixel in the image should then reflect the average 'complexity' of the subregion, with the more complex subimages (such as the ear segment) having more fine edges. The subimage with the largest mean value is selected as the correct ear segmentation.

Figure 17: Final Result of Segmentation

3.1.3 Reliability of Algorithm

In order to evaluate the consistence of this method, we applied it to three further test cases. In the four examples tested, two had the ear successfully isolated. A third had the ear poorly isolated, but did manage to capture a portion of the ear. A fourth failed to identify the ear entirely. This was most likely due to his highly textured shirt.

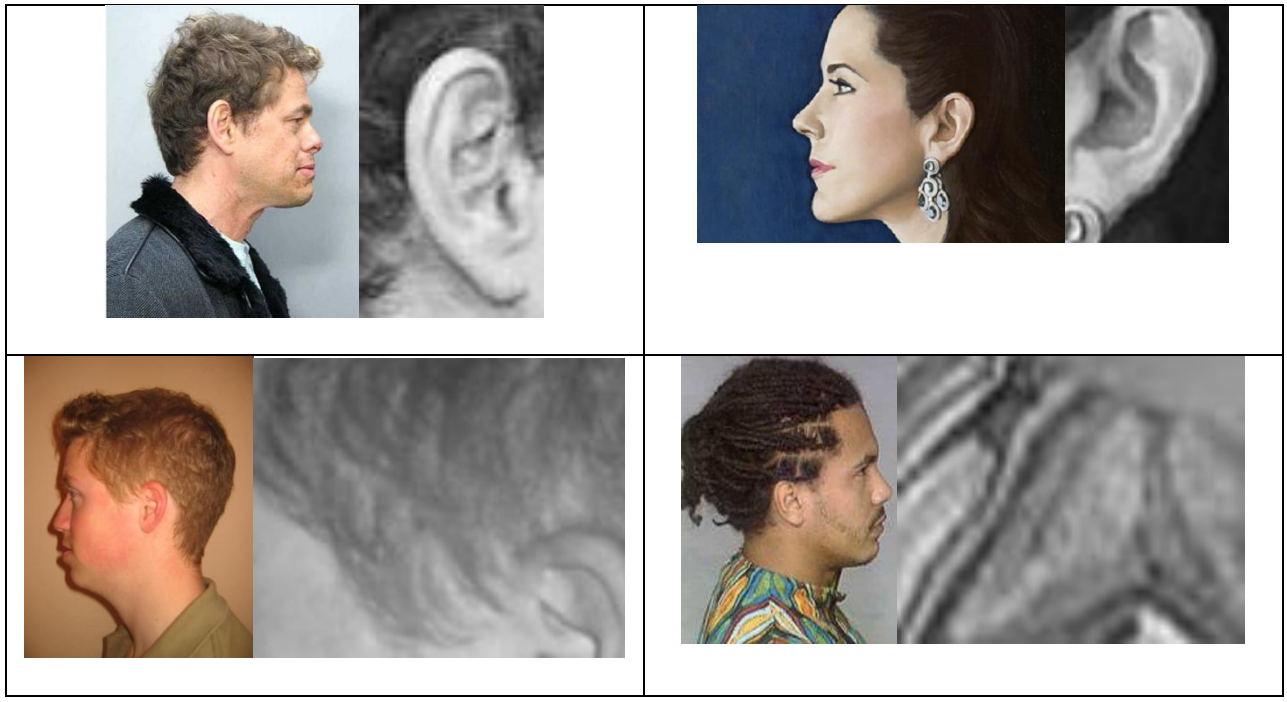


Figure 18: Various Test Results

3.2 Force Field Transform

3.2.1 Matrix Setup

The force field transform described by Hurley is a linear transformation, and as such has an equivalent matrix equation. For a simple 2x2 pixel image situation, the descriptive equations are described in the following.

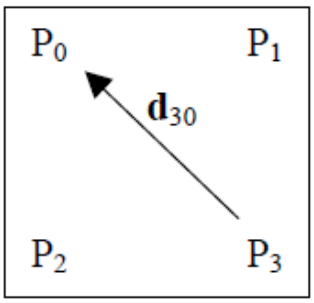


Figure 19: Four-Point Force Example

In the previous diagram, d_{30} represents the distance vector between P_3 and P_0 as determined by the inverse square law, given by:

$$\mathbf{d}_{ji} = \frac{\mathbf{r}_i - \mathbf{r}_j}{\left|\mathbf{r}_i - \mathbf{r}_j\right|^3}$$

Figure 20: Distance Vector for Force Field

The force at a point $F(r_0)$ is thus determined by the distance vector or each pixel paired with that pixel and the corresponding pixels' brightness values, or:

$$\mathbf{F}(\mathbf{r}_0) = \mathbf{d}_{01}P(\mathbf{r}_1) + \mathbf{d}_{02}P(\mathbf{r}_2) + \mathbf{d}_{03}P(\mathbf{r}_3)$$

Figure 21: Force at a Point

This equation can be re-written in matrix column and row vector form as follows:

$$\begin{pmatrix} 0 & \mathbf{d}_{01} & \mathbf{d}_{02} & \mathbf{d}_{03} \end{pmatrix} \begin{pmatrix} P(\mathbf{r}_0) \\ P(\mathbf{r}_1) \\ P(\mathbf{r}_2) \\ P(\mathbf{r}_3) \end{pmatrix} = \mathbf{d}_{01}P(\mathbf{r}_1) + \mathbf{d}_{02}P(\mathbf{r}_2) + \mathbf{d}_{03}P(\mathbf{r}_3) = \mathbf{F}(\mathbf{r}_0)$$

Figure 22: Vector Equation, Single Point

Concatenating on the vectors equations corresponding to every other point yields a full description of the force field in matrix equation form.

$$\begin{pmatrix} 0 & \mathbf{d}_{01} & \mathbf{d}_{02} & \mathbf{d}_{03} \\ \mathbf{d}_{10} & 0 & \mathbf{d}_{12} & \mathbf{d}_{13} \\ \mathbf{d}_{20} & \mathbf{d}_{21} & 0 & \mathbf{d}_{23} \\ \mathbf{d}_{30} & \mathbf{d}_{31} & \mathbf{d}_{32} & 0 \end{pmatrix} \begin{pmatrix} P(\mathbf{r}_0) \\ P(\mathbf{r}_1) \\ P(\mathbf{r}_2) \\ P(\mathbf{r}_3) \end{pmatrix} = \begin{pmatrix} \mathbf{F}(\mathbf{r}_0) \\ \mathbf{F}(\mathbf{r}_1) \\ \mathbf{F}(\mathbf{r}_2) \\ \mathbf{F}(\mathbf{r}_3) \end{pmatrix}$$

Figure 23: Force Field Equation

This application first generates the d-matrix based on the size of the test image, then calculates the force field based on the above equation. The d-matrix generation actually accounts for the majority of the processing time of this process, usually around ~80 to ~100 seconds. Applying the force field transform to our test image yielded the following results.

Figure 24: Starting Image

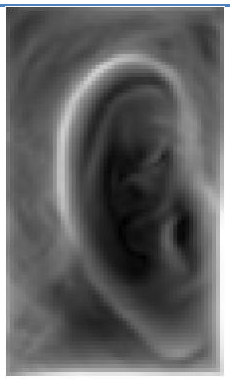


Figure 25: Force Field (Magnitudes)

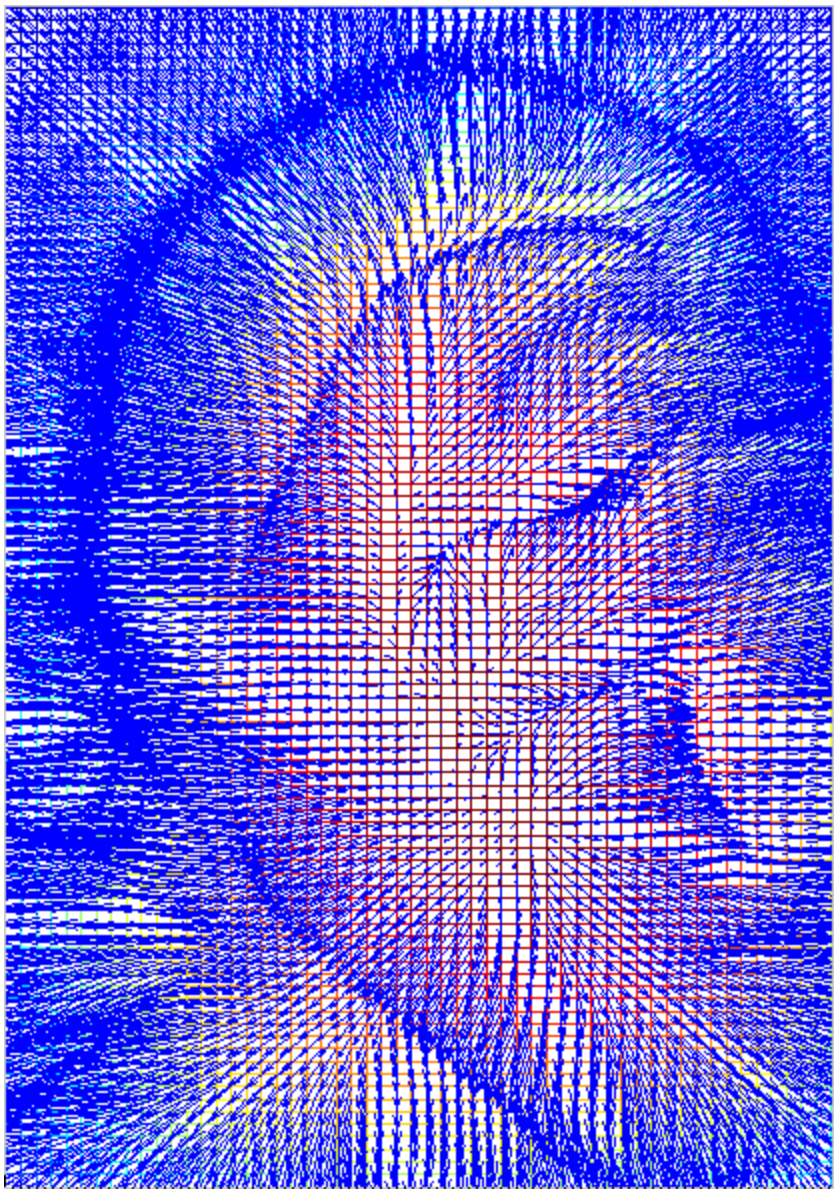


Figure 26: Force Vector Field of Test Image

After calculating the force field vectors for the image, we take the divergence of the force field in order to expose the channels and ridges of the field. Divergence field is then contrast normalized using histeq(). The contrast improved result is converted to a binary image, using a threshold of 190/255. To improve the template matching process in the next step, we also placed a 6px mask around the margins of the image in order to reduce the effects of spurious white pixels.

Figure 27: Divergence of Force Field

Figure 28: Contrast-Improved Divergence Field

Figure 29: After Binarization

The binary image that results from the previous processing is then compared against a template.

3.3 Rotation and Scale Invariant Template Matching

3.3.1 Circular Sampling Filter

To test the effectiveness of the template matching procedure, we used a slightly blurred and cleaned up version of the binary image resulting from the previous section as the template. This was compared against the actual output of the previous section. Before any sampling we apply a light Gaussian filter to both the template and the target image.

The circular sampling filter was chosen to operate at radii of [0 2 3 4 6 8 9 10 12 14 15] and at scales of [0.5 0.6 0.7 0.8 0.9 1.0 1.1]. These radii and scales were chosen based on the size of the template and the target image. In a more general situation these values would be scaled based on the relative sizes of the template and test images. The generated sampling filters were convolved with the template scaled by the factors specified above, and the value of the

centermost point of the template was noted after each. The result is s arrays of length r, where s is the number of scales tested and r is the number of radii test.

Figure 30: Several of the Cifi Filters Applied

The target image is sampled at its original scale using each of the r filters. This produces an x by y by r matrix of sampled values, or an r-length array per pixel. These two operations only take a fraction of a second to compute. The main time sink in this process is the next step, which correlates each of the template scale-sample arrays and the sampled target image.

For each pixel in the target image, the r-length array generated from sampling is correlated with each of the s arrays corresponding to the sampled template. The array with the best correlation corresponds to the probable scale of matching for that pixel. Both the best correlation and its associated probable scale are saved to two separate matrices (CisCorr_AQ and CisPS_AQ, respectively). This operation takes about 25 seconds for two 96x56 images.

Once complete, we search through the correlation matrix CisPS_AQ for correlation values above a threshold t1 = 0.80. A higher threshold improves computation time for each following candidate pruning process, but increases the likelihood of false negatives. The pixels that meet these selection criteria are marked as First Grade Candidate pixels, and are the basis upon which Radial Sampling is applied.

Figure 31: First Grade Candidate Pixels

3.3.2 Radial Sampling Filter

The radial sampling filters are defined to operate on a pixel at a radius equal to the largest radius used for the circular sampling filters, scaled by the probable scale at that pixel. They are set at discrete angular intervals of 5 degrees for the range of -180 to 180 degrees.

Figure 32: Some Rafi Filters

These filters (for a radius of 15 pixels) were first applied to the template image, with only the value of the center of the image being stored. This results in an array of length α, where α is the number of angles used for the sampling filters. The filters were then generated for each of the First Grade Candidate Pixels, with the radius equal to 15 pixels scaled by the probable scale at that pixel.

After the sampling processes, the sample array for each pixel is correlated against that of the template α times. For each comparison, the template's sample array is circularly shifted per comparison. The instance that provides the best correlation between the two corresponds to the probable rotation of template relative to the target image. Both the best correlation and the corresponding probable rotation for each pixel are stored off in two separate matrices.

Once correlations and probable rotations are calculated, the correlation matrix is searched through to select pixels with a correlation above a threshold $t_2 = 0.5$. These become the Second Grade Candidate pixels used within the following procedure.

Figure 33: Second Grade Candidate Pixels

3.3.3 Template Matching Sampling Filter

The last stage of the process uses an adjusted template image, which has been rotated and scaled to its most likely values for each Second Grade Candidate pixel. The size of this adjusted template image is uses as the bounding rectangle around its corresponding Second Grade Candidate Pixel, giving us two images of the same size to correlate against each other. Both of

these images are reshaped into a column vector, and their correlation is calculated. Of these points, those that result in a correlation value above a threshold $t_3 = 0.4$, are selected as Third Grade Candidates, which proceed to final processing. It is expected that, due to the blurring of both images before matching, there may be several points that could legitimately be the 'center' of the match.

Figure 34: Third Grade Candidate Pixels

Note that in Figure 34: Third Grade Candidate Pixels, all of the remaining candidate pixels are bunched up close to the center of the image, which is to be expected since the test and target images are near-identical to each other. In this particular case the best match should be the one that is closest to the center of the test image. However, this cannot be assumed for the general case, as the test images are not as idea as in this case. The point of matching is chosen by selecting the Third Grade Candidate pixel with the greatest correlation against the template. This point is thus representative of the quality of matching between the template and the target image.

It is then necessary to set a final threshold to determine whether or not the two images are actually a match. Although it is not applied here, it would simple process to apply this matching algorithm to a number of test cases involving both positive and negative matches, and then choose a threshold that minimizes both the number of false positives and the number of false negatives.

Figure 35: Point of Best Matching

In this test case, the best matching point had a correlation value of 0.4898.

4. Conclusions:

In this paper, we have presented a fairly complete ear biometric processing system. The key difficulties that were found during design and testing were related to the ear detection and segmentation and due to the time necessary to calculate the force field transform. The detection processes described here was fairly quick, but not quick enough to implement in real time. A quicker and likely more reliable detection system could probably be implemented using a

Cascaded Adaboost methodology, though this would require a large dataset for training, as well as a long amount of time for the training process.

An alternative to the force field transform may be to use the difference of erosion and dilation method used to enhance edges and contours in the detection phase, along with a topographic labeling method to mask away distracting elements. Switching to a combination of these methods would greatly reduce the amount of time required to process the images, as they do not require a substantial matrix setup time like the Force Field Transform requires.

The methods described here will work best for a direct, side view image of a person. A reasonable application of this system would be as part of an entrance hallway security system, that takes a profile view photo of people as they are forced to walk through a straight, narrow entrance. If the computation time for the process can be decreased, it may be possible for a biometric identification to be determined before the target reaches the end of the hallway.

References

Hurley, David J. “Force Field Feature Extraction for Ear Biometrics,” CiteSeer Beta. Web. 18 Nov. 2010 <http://eprints.ecs.soton.ac.uk/6792/1/ThesisDave22Mar02.pdf>

Hurley, David J., Mark S. Nixon, John N. Carter, “Force Field Energy Functionals for Ear Biometrics,” Computer Vision and Image Understanding, vol. 98, pp. 491-512, 2005. <http://portal.acm.org/citation.cfm?id=1077716>

Hurley, David J., Mark S. Nixon, John N. Carter, “Force Field Energy Functionals for Image Feature Extraction,” Image and Vision Computing, Volume 20, Issues 5-6, 15 April 2002, Pages 311-317, ISSN 0262-8856, DOI: 10.1016/S0262-8856(02)00003-3. <http://www.sciencedirect.com/science/article/B6V09-451NRD9-1/2/181d7d166d4a29e8d8ecc88cd7c00cc6>

Kim, Hae Yong and Sidnei Alves de Araujo. “Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness, and Contrast.” *Lecture Notes in Computer Science* 2007, Volume 4872/2007, 100-113. Web. 18 Nov. 2010 <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.8124&rep=rep1&type=pdf>.

Lankarany, Milad and Alireza Ahmadyfard. “Ear Segmentation Using Topographic Labels.” *Milad Lankarany.* Web. 18 Nov. 2010 <http://miladlankarany.synthasite.com/resources/VISAPP2009_Lankarany.pdf>.

Nixon, Mark S. et al. “On Using Physical Analogies for Feature and Shape Extraction in Computer Vision. ” *The Computer Journal* 2009. Web. 18 Nov. 2010 <http://comjnl.oxfordjournals.org/content/early/2009/08/07/comjnl.bxp070.abstract>.