5-2014

Ear Contour Detection and Modeling Using Statistical Shape Models

Satish Ravindran
Clemson University, satish.ravindran@gmail.com

Ear Contour Detection and Modeling using Statistical Shape Models

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Satish Ravindran
May 2014

Accepted by:
Dr. Ian Walker, Committee Chair
Dr. Damon Woodard
Dr. Stanley Birchfield

Abstract

Ear detection is an actively growing area of research because of its applications in human head tracking and biometric recognition. In head tracking, it is used to augment face detectors and to perform pose estimation. In biometric systems, it is used both as an independent modality and in multi-modal biometric recognition. The ear shape is the preferred feature used to perform detection because of its unique structure in both 2D color images and 3D range images. Ear shape models have also been used in literature to perform ear detection, but at a cost of a loss in information about the exact ear structure. In this thesis, we seek to address these issues in existing methods by a combination of techniques including Viola Jones Haar Cascades, Active Shape Models (ASM) and Dijkstra's shortest path algorithm to devise a shape model of the ear using geometric parameters and mark an accurate contour around the ear using only 2D color images. The Viola Jones Haar Cascades classifier is used to mark a rectangular region around the ear in a left side profile image. Then a set of key landmark points around the ear including the ear outer helix, the ear anti-helix and the ear center is extracted using the ASM. This set of landmarks is then fed into Dijkstra's shortest path algorithm which traces out the strongest edge between adjacent landmarks, to extract the entire ear outer contour, while maintaining a high computational efficiency.

Dedication

I would like to dedicate this thesis to my entire family for all the love and support they have given me, and to two very special people, my late grandfather MRM Nair and my late uncle Rajeev Menon for being absolutely wonderful human beings.

Acknowledgments

I would like to express my sincere gratitude towards my advisors Dr. Woodard and Dr. Birchfield, without whom this thesis wouldn't have been possible. Both their technical inputs, insights and guidance were instrumental in guiding me towards the completion of this work. I would also like to thank Dr. Walker for agreeing to be my committee chair and for his prompt feedback and support. I would also like to extend my gratitude to the entire Biometrics and Pattern Recognition Lab for all the help and brainstorming sessions during the research meetings. And last but not the least, I would like to thank my friends at Clemson; Priyanka, Deepika, Arijit, Sumod, Aman, Shubadha, Aesha and Keyur for making my life at Clemson memorable and a lot of fun.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Ear detection has been an active area of research in recent years due to a multitude of applications in two main fields: biometrics and head tracking. In the biometrics community, the ear has been used independently as a modality as well as a supplementary cue in multimodal biometric systems. For example, when used in conjunction with face recognition systems, it has been proven to improve the performance significantly [14]. In the tracking community, ear detection has been used with face detection to track head movements and to perform pose estimation [19], [26]. For a biological trait to qualify as a modality, it must satisfy four main constraints [13], [12].

- Universality: This means that it should be present in all the individuals. In biometrics, unlike other modalities like fingerprints, the ear is much less prone to wear and tear and so is mostly present in all individuals.

- Distinctiveness: This means that it should be unique to every individual. In biometrics research, this is an extremely important constraint since the modality needs to be uniquely associated with an individual. In tracking, though the constraint is not as strict, the object to be tracked should be distinctive enough to be able to extract features which distinguish it from the background. The ear performs extremely well in this regard. The ear shape is unique to the extent that even the left and the right ear of the same individual are not symmetric.

- Permanence: This means that the trait should be invariant over a long period of time. In biometrics research, it is important that the features don't change over time since while performing recognition, it could result in mismatches. As for tracking applications, it simply

means that it is easier to track a rigid object or an object that does not deform, than an object that undergoes large amounts of shape variation. The ear has been proven to be mostly invariant over a person's lifetime. The change in the size of the ear is much lesser compared to the rest of the body [25], [15].

- Collectability: This means that the trait should be quantifiable and easy to collect. For the ear, in spite of some issues with occlusion by hair and accessories, especially in women, collecting ear images is still much easier than getting iris scans or fingerprints, especially if the subject is not co-operative. This makes it extremely vital for security and forensic applications. In both biometric research and tracking, the features that are extracted should be measureable. The ear features are much easier to measure than the minutiae of fingerprints or iris features [1], [21].

According to Iannarelli, next to fingerprints, the ear has the most unique design, characteristic features and peculiarities for the purpose of identification [10]. The earliest known record of the ear being proposed as a viable biometric was given in 1890 by Bertillon [24], who wrote, 'The ear, thanks to these multiple small valleys and hills which furrow across it, is the most significant factor from the point of view of identification' [10]. As in a typical biometric system, the first and the foremost step to create an ear recognition system is ear detection. The accuracy of the whole system is extremely dependent on the accuracy of the detection step. Therefore, accurate ear detection is vital to improve the performance of the system. A typical ear recognition system is shown in Fig 1.1

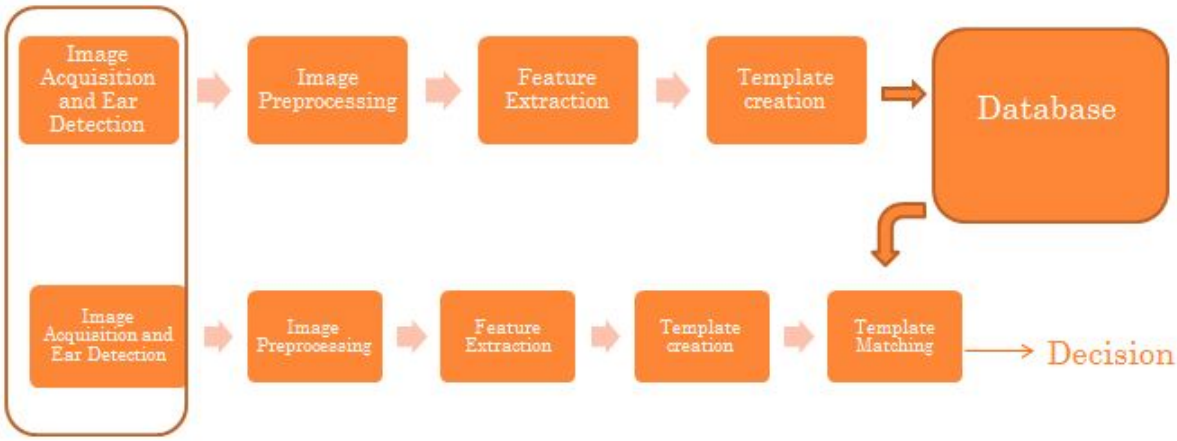

Figure 1.1: A typical ear recognition system

To understand the motivation behind this thesis, it is important to know a little bit about the structure of the ear. The outer ear anatomy consists of the outer helix, the antihelix, the lobe, the antitragus and the concha as shown in Fig 1.2. It is quite clear that the ear certainly has a very unique and distinctive shape. The main features used for ear based tracking and recognition are the helix, antihelix, the lobe and the ear center.

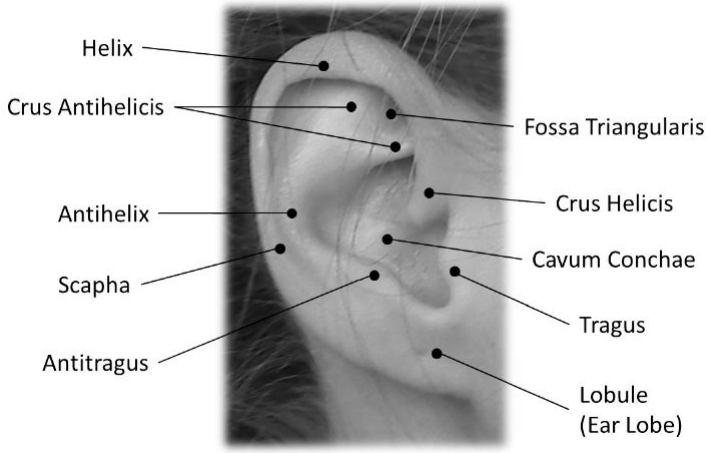

Figure 1.2: Structure of the ear[21]

1.0.1 Challenges in ear detection

There are many issues in formulating an accurate ear detection system. The most obvious ones are in the data collection stage, where it is difficult to even extract an image containing an ear. This could be because the ear is occluded by hair or accessories. Another one is where the ear is not visible because of the head orientation of the subject. This thesis does not deal with these issues. We assume that the ear is fully visible in the image and the pose does not change too much. This thesis deals with two other challenges in ear detection: devising a robust geometrical model for the ear and detecting an accurate contour around the ear. As will be explained in the next chapter, most of the current segmentation techniques don't address these two issues. The importance of having an ear shape model and an accurate ear contour is explained in the next section.

1.1 Motivation

Though ear detection has been an area of research for quite a long time, it is only in recent years that ear shape has been used as a feature. Traditionally, color and texture were the primary features which were used to perform detection. But there are inherent problems in using these features for the ear since there is not much color variation in the ear region. The shape of the ear, on the other hand, has been proven to be extremely unique. Currently, there are two kinds of ear shape features that can be leveraged to perform detection.

- 2D Shapes: The ear shape in RGB images consists of the contours caused by the high gradient variation due to depth changes. The ear helix and the anti-helix are the most prominent edges in a 2D image. Both are very distinctive curves whose shape properties can be used to perform detection and recognition.

- 3D Shapes: The shapes in 3D images consists of surface curvatures formed by the hills and ridges in the ear. The 3D surface of the ear contains a lot more information which can be used to perform recognition and with recent improvements in 3D sensors, 3D shape based detection is generating a lot of interest in the tracking and biometrics community. The importance of an accurate contour around the ear can be gauged by the fact that the Iterative Closest Point algorithm (ICP), the favored algorithm for 3D ear recognition is extremely sensitive to outliers. Most of the algorithms which use ICP tend to do some preprocessing to eliminate outliers.

Another very active area of research is in ear modeling. There are many reasons to devise a ear model. It reduces the feature vector dimensions used to represent the ear. It can be used to perform coarse biometric recognition and quickly eliminate non-ears or false matches in a sliding window detection scheme. A model can also be devised which can extract the important features of the ear like the ear helix, anti-helix and the ear center in constant time. A model enables a top down approach to detection where global constraints can be used to offset local errors.

Inherently, a model and an accurate ear contour are difficult to achieve simultaneously since they both have different objectives. A model attempts to capture the maximum information possible using the fewest possible parameters. An accurate ear contour depends on the fine and local details present around the ear edges. But since both are important in a modern detection scenario, this thesis is an attempt to devise a parametric model for the ear using statistical models while simultaneously

extracting an accurate ear contour in a computationally efficient manner. The model that we have chosen to represent ears is Active Shape Models (ASM). Using Active Shape Models enables us to control the feature points and devise a comprehensive and robust geometric model for the ear. Every ear in an ASM based model can be described by a set of eigenvectors and eigenvalues. The parametric model also enables us to segment out the different features like the ear outer helix, anti-helix and the ear center without any additional processing. This could be very useful in part based detection and recognition schemes. Using parametric models to approximate the shape has the drawback that some of the fine details of the exact shape will be lost. Therefore, this thesis also describes a way to improve the accuracy of the shape models by augmenting it with an interpolation technique resulting in a more accurate contour around the ear boundary.

1.2 Layout of the thesis

This thesis is divided into three main chapters. In the next chapter, we will give a brief introduction about the history of ear detection focusing mainly on the shape based techniques and explain some of the issues in the current techniques which are addressed by our method. Then in chapter 3, we will proceed with the background theory and the underlying concepts behind the technique proposed in this thesis. Finally in the last two chapters we will present the results, the future work and the conclusions drawn from the experiments performed by us.

Chapter 2

Related Work

We saw in the previous chapter how the ear shape is an extremely unique and important feature and the need for an accurate ear contour detector and shape model in modern applications. In this chapter we will describe some of the ear detection techniques present in literature with special focus on the shape based detection and recognition methods. We will cover both 2D and 3D detection techniques, though in this thesis, the scope is limited to 2D ear shape detection.

2.1 Hough Transform

A paper by Arbab-Zavar and Nixon described a method which used an elliptical Hough transform to model the ear [3]. This was based on the basic premise that the ear is almost an ellipse and can be detected by using elliptical parameters. This technique had a huge advantage that it also gave a model for the ear in terms of the ellipse parameters, which could then be used for recognition. Some sample results are shown in Fig 2.1.

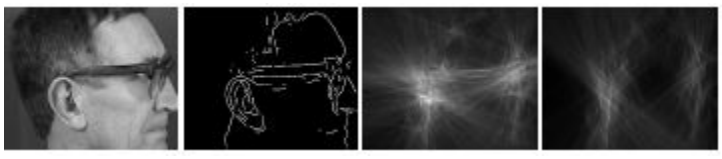

Figure 2.1: Ear detection using an elliptical Hough Transform

They achieved an enrolment rate of 100% on the XM2VTS database and 91% on the UND

database. While the detection rate is really great, it does not accurately capture the exact shape of the ear and a lot of information about the exact structure is lost.

2.2 Image Ray Transform

In this paper, Cummings et al. proposed an ear detection technique based on an analogy to light rays [8]. Here, the image is modeled as a medium with every pixel having a refractive index proportional to its intensity. Then thousands of light rays are simulated at random locations around the image and a path is traced out as the ray passes from one medium to another following the laws of optics. Every time a ray enters a pixel, an accumulator array is incremented. The idea was that all the tubular features, like the ear helix, would act as a waveguide and more light rays would enter the structure and remain within it, giving rise to a higher value in the accumulator array. They achieved a 99% detection rate on the XM2VTS database with 252 images. Some sample results are shown in Fig 2.2. While the detection rate is very impressive, it does not always detect the whole ear and it is also prone to errors when the subject is wearing spectacles and other things. Also, the XM2VTS database is of a very high quality with the images taken against a uniform blue background. So it remains to be seen how effective this technique will be in more natural conditions.

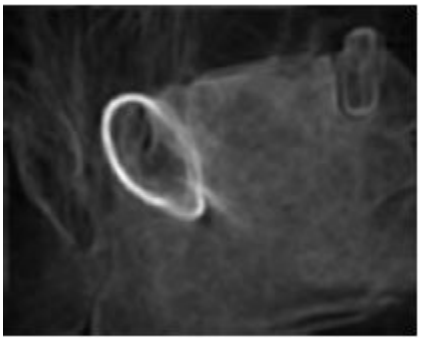

Figure 2.2: Ear detection using Image Ray Transform

2.3 Active Contours

Active Contours are deformable models which operate under internal and external forces to fit object boundaries. Since ears have a very strong gradient around the helix, Yan and Bowyer devised an approach to fit an active contour around the ear boundary [29]. The first step was

detecting the ear pit by first detecting the nose tip, then using skin detection to segment out the face and the ear and then doing surface curvature estimation followed by segmentation and classification. Once the ear pit is detected, an active contour algorithm is initialized with the ear pit as the starting point. The contour operates under the influence of the internal and external forces and moves towards the ear boundary. Some sample results on 2D images are shown in Fig 2.3.

Figure 2.3: Ear detection using Active Contours on RGB images

Since the segmentation on 2D images was not satisfactory, they used a combination of both the depth image and the 2D image to restrict the contour to the ear boundary. The results are shown in Fig 2.4.

Figure 2.4: Ear detection using active contours on RGB and depth images

Using only the depth image, they achieved a detection rate of 85% and using only the color image, they achieved a detection rate of 79%. But combining both the depth and color information improves the detection rate to almost a 100%. Once again, while the detection rate is very impressive, it requires both 3D and 2D data to achieve this rate.

2.4 Outer Helix Shape Based Detection

Ansari and Gupta used a combination of edge detection and curvature estimation to crop out the outer boundary of the ear [2]. After edge detection on a side face image, the edges are segmented into concave and convex edges. Then to localize possible helix curves, certain conditions are imposed. A curve is said to be a candidate helix if it has some curve running parallel to it within a certain distance. After this certain angular constraints are used to detect the correct ear helix. A sample result is shown in Fig 2.5.

Figure 2.5: Ear detection using outer helix

They achieved a localization accuracy of 93.3% on 700 images on the IIT-K database. But this method is extremely sensitive to the threshold parameter, which decides the distance between the ear outer and inner helix.

2.5 3D Model Based Detection

Chen and Bhanu devised a model based ear detection scheme for 3D images [5]. The model they used was a set of 3D (x,y,z) co-ordinates of landmark points around the ear helix and anti-helix. The step edge magnitude in a side profile image was computed to find a set of candidate pixels for the ear helix and anti-helix. They made use of the fact that in a depth image, there will be lots of discontinuities around the ear because of its valleys and hills. A step edge is an edge which is caused by large depth discontinuities, rather than by illumination or color changes. They detected the ear outer and inner helix as shown in Fig 2.6. Then dilation and connected components labelling was used to divide the edge pixels into clusters, with each cluster considered as a candidate ear. Then using a modified Iterative Closest Point algorithm, the mean square error was computed between a model ear built from a person and the set of clusters. The cluster which gave the least error was

designated as the correct ear. They achieved a 92.6% detection rate on 312 range images. A sample result is shown in Fig 2.6

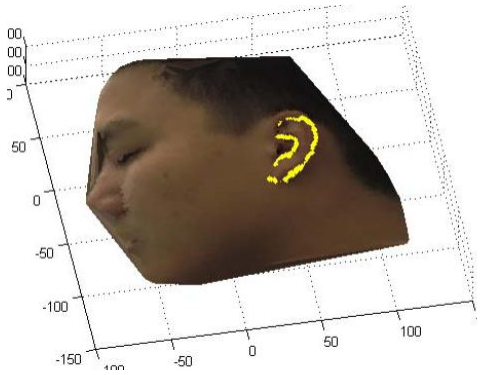

Figure 2.6: Ear detection using step edges

2.6 Histogram of Categorized Shapes

This technique was developed by Zhou et al. for 3D ear detection [31]. Here they compute features which capture the 3D features like the saddle and ridges around the ear helix and anti-helix, and the hills and valleys in the inner ear region. These features are parameterized using shape index and curvedness values. Then using a sliding window approach and a Support Vector Machine classifier, the ear is segmented as shown in Fig 2.7. They achieved a detection rate of 100% on 142 range images. This was also one of the most computationally efficient methods to detect ears in 3D images. The average processing time per image was 2.8 seconds.

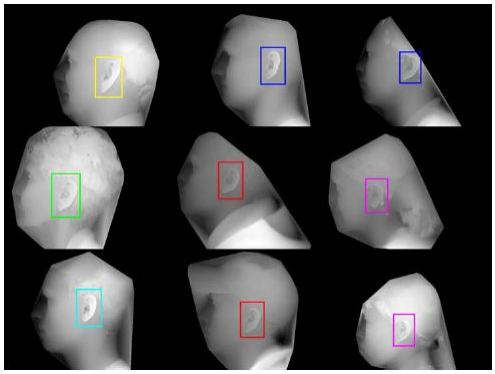

Figure 2.7: Ear detection using Histogram of Categorized Shapes

2.7 Graph Connected Components

In 2012 Prakash and Gupta devised a 3D ear detection scheme based on connectivity graphs [22]. The edges are extracted on a range image to get a list of the possible candidates for the ear contours. The outliers are pruned and a connectivity graph is created. A pre-computed shape distribution created from a training set is then compared with a candidate list and a match is found. A sample result is shown in Fig 2.8.

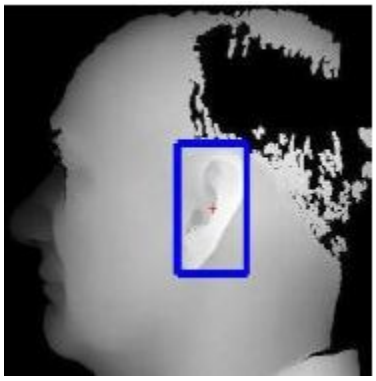

Figure 2.8: Ear detection using Connected Components in 3D images

2.8 Problems with current ear detection techniques

As mentioned before, all the techniques described until now either come up with an accurate ear boundary [2], [8], [29] or have a model based detection scheme [3], [5], [31]. And except for [31], which works only on range images, the rest of the models are very weak approximations of the actual shape. A lot of information is lost about the actual structure of the ear. Both the shape model and an accurate ear contour are very essential in modern applications as was explained in Chapter 1, where we described the motivation behind this thesis. In the next section we will describe the technical and the implementation details of the combination of techniques we used to address both the aforementioned issues.

Chapter 3

Approach: Ear Contour Detection and Modeling

We have seen that the techniques discussed in the previous chapter have either one of the two following shortcomings; they either detect the ear contour accurately, or they develop a geometrical model for the ear. The importance of having a shape model and an accurate ear contour has already been discussed at length in the last two chapters. In this chapter, a technique is described which addresses both these issues. Our approach uses a combination of different algorithms to implement a fully automatic ear contour detection and shape modeling system. In a nutshell, the Viola Jones Haar Cascades classifier is used to get a bounding box around the ear. Then Active Shape Models are used to extract distinguishing landmark points around the ear. The output of the Active Shape Model is then fed into a modified form of Dijkstra's shortest path algorithm which draws a smooth contour around the ear outer boundary. In the subsequent sections, the underlying concepts and the implementation details of the methods used in this thesis are explained.

3.1 Viola Jones Haar Cascades

In a landmark paper in 2001, Viola Jones devised one of the most popular supervised learning algorithms to detect objects [27]. It was based on an analogy of Haar wavelets which captures structural information about the object. To select relevant Haar features, Adaboost was

used after which to improve the computation time, a cascade of classifiers was used to eliminate non-ears quickly. They have been successfully used to detect a wide range of items, the most popular being the face detector.

3.1.1 Haar features

Haar features are similar to Haar wavelets which mimic the patterns interpreted by the human visual system. Just like any other filter, they extract structures which look similar to the filter from an image. There are different types of features as shown in Fig 3.1, each corresponding to a particular structure in the image.

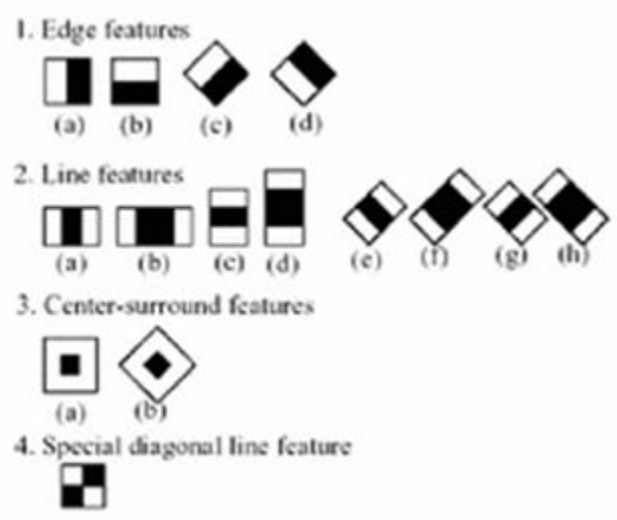

Figure 3.1: Some standard Haar features used in Viola Jones[27]

The different structures, orientations and scales of the features capture different kinds of structural information within an image window. A filter response is a just single scalar which is the difference of the sum of the pixel intensities under the bright elements and the sum of the pixel intensities under the dark elements of the filter. An example structure element evaluated at the ear center is shown in Fig 3.2. But since there are a large number of permutations possible for these features, evaluating these filter responses is extremely expensive computationally. To overcome this problem, Viola Jones used the concept of integral images.

Figure 3.2: An example of a Haar feature evaluated at the center of the ear

3.1.2 Integral image

Integral images have been used extensively in the vision community to quickly compute rectangular filter responses. The integral image computation is a pre-processing step which takes $O(mn)$ time, where m and n are the width and height of the image. The entire Integral image can be computed by a single pass through the image using dynamic programming. The value at (x, y) for the integral image is given by Eq 3.1

$$I(x, y) = i(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1) \tag{3.1}$$

where $i(x, y)$ is the grayscale pixel value of the original image at (x, y). Therefore, every pixel in an integral image contains the sum of the intensity values of all pixels to the upper left region of that pixel. It is possible to know the total intensity of any rectangle of any size in constant time using only four lookups.

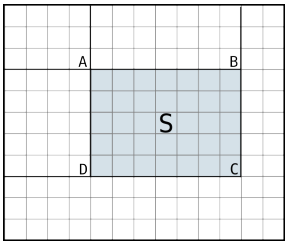

Figure 3.3: Computing the area using integral image[23]

As shown in Fig 3.3, the sum of the shaded area can be computed by Eq 3.2

$$Sum = I(C) + I(A) - I(B) - I(D) \tag{3.2}$$

Thus for the different rectangular Haar filters, to compute the total intensity of the area under the bright or the dark pixels, only a constant amount of lookups will be required irrespective of the filter size.

3.1.3 Adaboost

Even with an improved computation time for the filter responses, there are still thousands of Haar features that need to be evaluated. But not all features will be relevant for ear detection. Therefore it is important to select features which actually capture useful structural information about the ear. Adaboost is a machine learning algorithm which is used to create a strong classifier by combining a bunch of weak classifiers A weak classifier is a classifier which performs better than random guessing, which means that it correctly detects the ear more than 50% of the time. Every weak classifier uses a single Haar feature, evaluates a filter response and gives a binary output based on the response. The output of a weak classifier is given by Eq 3.3

$$\begin{aligned} f_i &= Sum(r_{i,white}) - Sum(r_{i,black}) \\ h_i &= \begin{cases} 1 & if\ f_i > threshold \\ -1 & if\ f_i \leq threshold \end{cases} \end{aligned} \tag{3.3}$$

where h_i is the decision, f_i is the response of the i^{th} classifier, $r_{i,white}$ and $r_{i,black}$ are the sum of the areas under the white filter elements and the black filter elements.

Adaboost helps in selecting the best features out of the huge number of features available. A feature is included if it detects the object correctly more than 50% of the time. It then uses a linear combination of weighted classifiers, where the weight is the misclassified error, to come up with the final output. A visual representation of the algorithm is depicted in Fig 3.4.

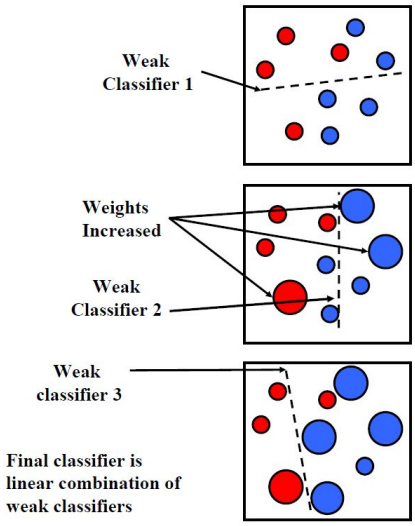

Figure 3.4: Adaboost overview[9]

Adaboost Algorithm [9]

1. The input is a set of training images $(x_1, y_1)...(x_n, y_n)$ where y_i is a flag to indicate the positive and negative examples.

2. Initially the weights are distributed uniformly as $w_{1,i}=\frac{1}{2m}, \frac{1}{2n}$ for $y_i = 0, 1$ where m and n are number of positive and negative examples respectively.

3. For $t = 1, ...T$, where T is the number of features required

   (a) Normalize the weights

   $$w_{t,i} \leftarrow \sum_{j=1}^{n} (w_{i,j})$$

   (b) For each feature j train a classifier h_j. The error, $\epsilon_j=\sum_i (w_i(h_j(x_i) - y_i))$. This gives a weighted error of the misclassification using the classifier h_j.

   (c) Choose the classifier h_t with the lowest error ϵ_t

   (d) Update the weights

   $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$

   where e_i=0 if x_i is classified correctly, e_i=1 otherwise and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

4. The final strong classifier gives a binary output depending on the conditions described below.

$$h(x) = \begin{matrix} 1 & if \ \sum_{t=1}^{T} \alpha_t h_t(x) \geq \ \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{matrix}$$

where $\alpha_t = log \frac{1}{\beta_t}$

3.1.4 Cascading

In any sliding window detection scenario, using the strong classifier at every sub window is a very expensive process, since there will definitely be more non-ears than ears. Therefore to improve the computation time, a cascaded approach is used where every stage consists of collection of weak classifiers together forming a strong classifier. The early stages are chosen such that they have a weak threshold for a positive detection. Therefore anything which does not look like an ear at all will be rejected by these stages. This way, the non-ears are eliminated quickly, with the threshold for detection becoming stricter at every stage. Any window which successfully goes through the whole cascade has a very high probability of being an ear. This process leads to a vast improvement in computational efficiency since all the classifiers do not operate on all the windows. A simple flowchart illustrating the algorithm is shown in Fig 3.5.

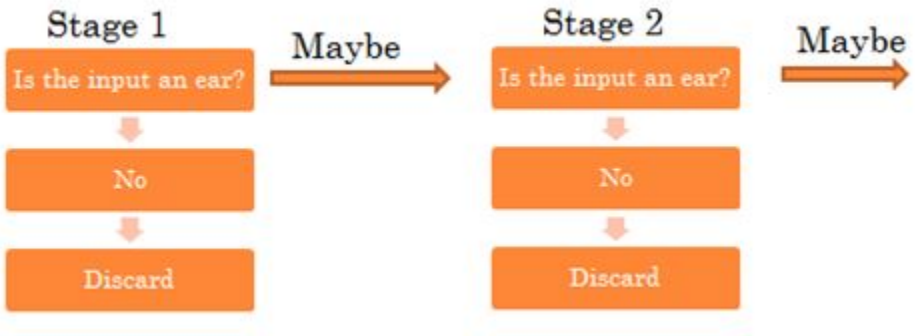

Figure 3.5: Cascading flowchart

The main challenge in implementing the cascading algorithm is to figure out the number of stages in the cascade, the set of features required for every stage and the threshold for every stage. Viola and Jones came up with a heuristic algorithm to implement the cascading stage.

Cascading Algorithm [9]

Input: FP_i is the maximum false positive rate/stage, TP_i is the minimum true positive rate/stage, F_{final} is the overall false positive rate

while $F_{current} \geq F_{final}$ **do**
  Add new stage
  while FP_i,TP_i *for this stage is not achieved* **do**
    Add features and retrain using adaboost
  end
end

Algorithm 1: Cascading for Viola Jones

3.1.5 Implementation description

The VJ detector that was used for this thesis has been described in [27]. OpenCv already had trained features for eyes, nose and mouth. Recently, features for the ear were also added. The next section briefly describes the implementation details of the trained Haar features for ear detection as described by [4]

Training

The authors of [4] used the FERET dataset to do the training and evaluation. They trained the VJ detector to detect ears in profile or almost profile images .The classifier was created with 5000 positive images and 15000 negative samples. The negative samples were basically images of wallpapers that did not contain the target pattern. Three different sets of classifiers were created, one to detect left ears, one to detect right ears and one to detect a generic ear. But for our experiments we used only the left ear classifier.

Evaluation

In the evaluation phase, they created a test set from a different subset of the FERET database and manually annotated the ear. A detection was classified as true positive if the detected window had at least a 50% overlap with the ground truth or if the detected window center was within a distance of $0.25 * w_g$, where w_g is the width of the ground truth window. The sliding window size used for every feature was 12X20 and it used 20 stages in the cascading step. Some sample outputs are shown in Fig 3.6

Figure 3.6: Sample outputs from the Viola Jones Haar Cascades detector

3.2 Active Shape Models

The output of the Haar Cascades stage is a rectangular window marked around the ear. It contains no information about the shape or geometry of the ear. The next step in our algorithm is to devise a shape model and extract the exact ear contour. Active Shape Models (ASM) are derived from Point Distribution Models (PDM), which represents the mean geometry of an object. The PDM is built from a set of training shapes and principal component analysis is used to determine a set of parameters to represent the shape.

ASMs are statistical distribution models where the landmark points undergo local displacements based on image evidence while confirming to an overall shape structure determined by the shape model. They have been extensively used in the past to model faces, mechanical assemblies and medical images [7], and more recently to normalize ears by modeling the ear outer helix [30]. It consists of two sub models: $The\,Profile\,Model$ and $The\,Shape\,Model$. The profile model consists of a gradient vector and a gradient covariance matrix for each landmark point. The shape model consists a set of eigen vectors describing the geometric correlation between the landmark points and a set of eigen values which define the allowable deformations that the model can undergo.

3.2.1 Building the model

The model is built by manually annotating the landmark points in the training image. Landmark points are distinctive features like corners or edges in the ear. A sample template of the annotated ear is shown in Fig 3.7

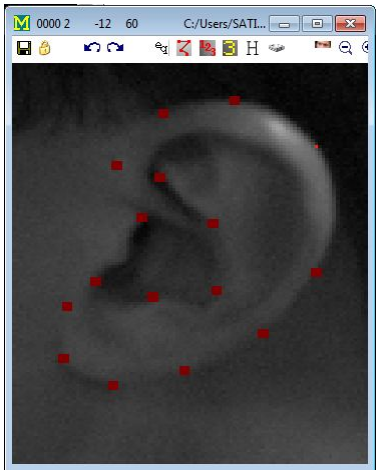

Figure 3.7: A sample annotated ear

Once the points have been annotated, the next step is to bring all the points in all the training images into a common co-ordinate frame. This is important since the model is attempting to capture the shape and the geometrical structure of the object. The position and scale of the object is not important. An iterative algorithm was used for this [7].

Algorithm to align shapes

1. Choose the first shape as the reference shape.
2. Translate the shapes so its centroid is at the origin.
3. Scale the reference shape such that $|x| = 1$.
4. Align all shapes with the mean shape as reference.
5. Recompute the mean shape by taking the mean of the current set of shapes.
6. Scale the updated mean shape such that $|x| = 1$.
7. Repeat until convergence.

3.2.1.1 The profile model

The profile model suggests local displacements for every landmark point, based on the image properties around the current landmark. Every landmark point is associated with a profile vector,

which is a sample of the normalized gradient along the normal to the model boundary at that point as shown in Fig 3.8 .

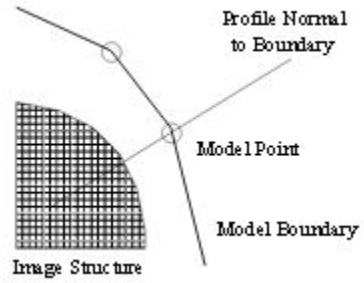

Figure 3.8: The profile model [6]

A sample image showing the profile models at some of the landmark points is shown in Fig 3.9

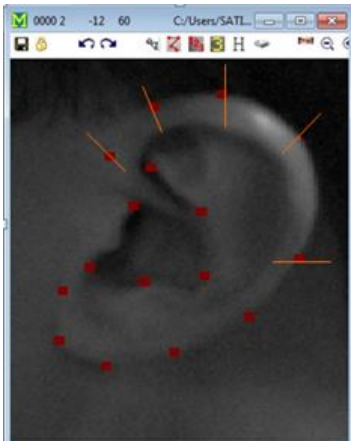

Figure 3.9: The profile model marked around the landmarks

In other words, the profile element at each position i along the profile vector, is replaced with the difference in gray level intensity between it and the element at $i-1$, and then normalized. The normalized gradient is used instead of the absolute gray values to offset the effect of illumination changes. During training, the profile model is built by creating a mean profile vector by averaging the profile vector values at the corresponding landmarks for all the training images. Then a covariance matrix is created using the profile vectors of all the landmarks of the entire training set. Every landmark is modeled as a Gaussian with a mean profile vector and a covariance matrix associated

with it. These two together make up the profile model. The profile model suggests new positions for every landmark at every iteration. These are calculated at displacements of ±3 along the normal to the model boundary. As shown in the figure above, the new suggested position is the one which is most similar to the model profile at that landmark. The Mahalanobis distance D is used to compute the similarity between a profile vector g and the corresponding mean profile vector $\bar{g}$ and the covariance matrix at that landmark S_g .

$$D = (g - \bar{g})^T S_g^{-1} (g - \bar{g})$$

3.2.1.2 The shape model

The shape model constricts shape suggested by the profile model to the shape of the ear and ensures that the shape is not distorted beyond recognition. It is represented by the following equation.

$$\hat{x} = \bar{x} + \phi b$$

where $\hat{x}$ is the final shape vector with the following format

$$\begin{bmatrix} x_1 \\ . \\ . \\ x_n \\ y_1 \\ . \\ . \\ y_n \end{bmatrix}$$

where x_1,y_1. . . x_n,y_n are the n co-ordinates of the landmark points marked around the ear, n is the number of landmark points in the model.

Also, $\bar{x}$ is the mean shape generated by the training set,

$$\bar{x} = \frac{1}{n_{shapes}} \sum_{i}^{n_{shapes}} x_i$$

where n_{shapes} is the number of training examples.

And ϕ is the matrix of the eigenvectors of the covariance matrix S_s of the training shape points.

$$S_s = \frac{1}{n_{shapes} - 1} \sum_{i=1}^{n_{shapes}} (x_i - \bar{x})(x_i - \bar{x})^T$$

The eigenvectors are sorted according to the corresponding eigenvalues and only those eigenvectors corresponding to the largest eigenvalues are used. Each eigenvalue measures the variance present along the corresponding eigenvector.

ϕ gives the principal directions in which the data is distributed. The final shape is given as a weighted sum of the principal components and the mean shape. b is the set of weights which controls how much weight to assign to a particular eigenvector. By keeping the limits of b within a range of $-3\sqrt{\lambda}$ to $+3\sqrt{\lambda}$ it is ensured that the model confirms to the shape of the ear. Now as mentioned above, the algorithm iteratively uses both the profile and the shape model to come up with a best fit for the ear. The profile model suggests local displacements and the shape model ensures that the model restricts the deformations so that the overall shape confirms to the shape of the ear.

3.2.2 ASM algorithm description

This is an optimization problem where we have a set of global constraints imposed by the shape model and a suggested displacement of the model points given by the profile model. The goal is to find the closest shape in the model space, defined by its eigenvectors, of the shape suggested by the profile model. This shape exists in the image co-ordinates system.

Multi resolution search

To enable the algorithm to converge faster while searching for the ideal fit, the search process is carried out at different scales. Here a Gaussian pyramid is used with each subsequent level obtained by smoothing the previous level and subsampling the result. Though the number of pixels is reduced by half in each dimension at every level, the profile vector length is still kept as same. When searching within a level, if the best fit is found within half of the profile length for more than 90% of the landmark points, then the search moves on to the next level and the whole process is repeated until we reach the final resolution.

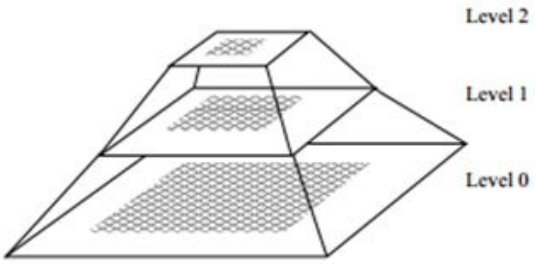

Figure 3.10: Multi-resolution search[17]

ASM search algorithm

Input: The start shape which is the mean shape around the initial position

Output: The final shape fitted around the object

repeat

  for *every model point* **do**

    for *a displacement of* ± 3 *along the gradient at that point* **do**

      1. Compute the profile model at different offsets.
      2. Compute the distance between the computed profile model and the mean model of the corresponding point obtained during training.
      3. Shift the point to the offset with the least error.

    end

  end

  Impose the global constraints of the shape model so that the shape does not get distorted.

until *convergence*;

Algorithm 2: Algorithm for ASM search

Fitting the profile model to the shape model

At every iteration, the profile model suggests a new position for every landmark point. The shape model then applies the global constraints to ensure that the shape is not distorted. An iterative algorithm is described to fit the shape model parameters to the shape suggested by the profile model [7].

Algorithm

Initialize the shape to the mean shape.

repeat

1. Align the model points to the points suggested by the profile model. $\hat{x} = \bar{x} + Pb$
2. The suggested shape by the profile model Y is in the image co-ordinate frame. To convert this into the model co-ordinate frame, the inverse of the transformation matrix from the previous step is used.

$$y = T^{-1}_{X_t, Y_t, s, \theta}(Y)$$

3. Project y into the tangent plane to $\bar{x}$ by scaling: $y = \frac{y}{(y \cdot \bar{x})}$
4. Update the model parameters to match to y

$$b = P^T(y - \bar{x})$$

5. Truncate b wherever necessary

until *convergence*;

Algorithm 3: Fitting suggested shape to ear model

3.2.3 Implementation details

Once the Viola Jones detector has cropped out the ear from the profile image, the next step is to fit the ASM to the ear. The basic theory behind the Active Shape Model has already been described in the previous section. For our experiments we used an ASM library called the Stacked Active Shape Model (STASM) [16], [17].

Training

The training set consisted of 82 images picked from the UND-Class J-2 database. All the images were of the left side profiles of different subjects. This was to ensure that the training set adequately captured the variation present in the database. A number of different locations for the landmarks were tried, but the best fit seemed to come from choosing 16 points around the ear marked as shown in Fig 3.7. The 16 key points consisted of 1 ear center, 9 points around the ear helix, 4 points around

the ear anti-helix and 2 additional points around the ear outer contour. The points are normalized with respect to the center of the image.

Testing

The testing was done on the 331 images for which Viola Jones correctly detected the rectangular region around the ear. Some of the sample outputs for ASM is shown in Fig.3.11.

Figure 3.11: Sample outputs from ASM

3.3 Interpolation using Dijkstra's Algorithm

The output of the ASM is a set of landmark points around the ear contour with the adjacent points connected by straight edges. While the ASM provides an excellent geometrical model for the ear, the contour detected by the model is not very accurate. There is a need to interpolate between the landmark points which accurately captures the ear contour as shown in Fig 3.12.

Figure 3.12: ASM output depicting the issues with the contour detection by the model

There are two ways this problem can be solved. One is the heuristic approach, where we keep approximate a curve by increasing the number of landmark points until a certain degree of accuracy is reached. This would involve clicking a lot of points manually and it would also increase the computational complexity for the ASM algorithm. The other way is to use Dijkstra's algorithm which automatically adapts to the image evidence and traces the strongest edge between two pixels. This leverages the fact that the ear has very strong gradients along the outer contour and in most

cases, the strongest edge between two points on the contour will be along the contour. In the vision community, Dijkstras algorithm has been used before to find the strongest edge between two pixels [18]. Here the problem is modeled as one of finding the lowest cost path between two nodes in a weighted graph. The image is modeled as a graph with the pixels representing the nodes which are connected by edges with its 8 neighbors as shown in Fig 3.13

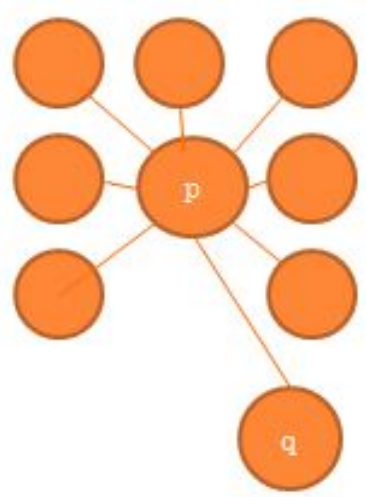

Figure 3.13: A pixel connected by weighted edges to its 8 neighbors

In the next section, a detailed description of the algorithm is given.

3.3.1 Local costs

The local cost of an edge between pixels p and q is defined by $l(p, q)$ which is the weighted sum of the following features.

Feature	***Symbol***
Canny Edges	l_c
Gradient Magnitude	l_g
Gradient Direction	l_d

Table 3.1: Local costs used in Dijkstra's algorithm

A linear combination of the features gives the total cost of an edge between the two pixels

as shown in Eq 3.4

$$l(\mathbf{x}_1, \mathbf{x}_2) = w_c l_c(\mathbf{x}_2) + w_g l_g(\mathbf{x}_2) + w_d l_d(\mathbf{x}_1, \mathbf{x}_2) \tag{3.4}$$

w_c, w_g and w_d are the weights assigned to the individual features. In our implementation, we have assigned equal weights to all the parameters. The Canny edge detector is used to give a higher weight to strong edges. But since the output of an edge detector is a binary image, the information about how strong the edge is at the pixel is lost. Therefore, the Gradient Magnitude cost which gives an indication of the strength of the edge at that point is also included. We need to assign a low cost to the strong edges. To do this, the gradient magnitude is normalized and inverted. Therefore, the cost of an edge is inversely proportional to the strength of the gradient at that point. The gradient direction is another parameter which is used to smooth the curve by assigning a high cost to sharp contours. If there is a large difference in the gradient direction between two pixels then a high cost is assigned to it. The gradient direction cost is also normalized.

$$\begin{aligned} l_c(\mathbf{x}_i) &= \begin{cases} 0 & if\ canny(\mathbf{x}_i) = 1 \\ 1 & if\ canny(\mathbf{x}_i) = 0 \end{cases} \\ \mathbf{G} &= \sqrt{{\mathbf{I}_x}^2 + {\mathbf{I}_y}^2} \\ l_g &= 1 - \frac{G}{max(G)} \\ l_d(x_1, x_2) &= \frac{|arctangent(\frac{I_{y1}}{I_{x1}}) - arctangent(\frac{I_{y2}}{I_{x2}})|}{2\pi} \end{aligned} \tag{3.5}$$

where I_x and I_y are the gradients in the **x** and **y** directions, **G** is the gradient magnitude at (**x**,**y**)

3.3.2 Implementation details

The output from the ASM is fed into the Dijkstra's algorithm in a sequence. The standard Dijkstra's algorithm traces the shortest path between a node and every other node in the graph. We stop the search when the destination pixel is reached. The strongest edge is marked between adjacent landmark points as shown in Fig 3.14

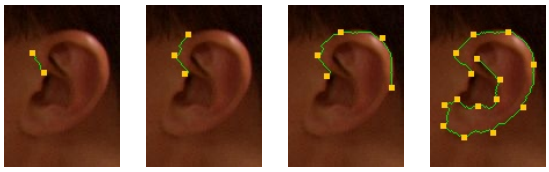

Figure 3.14: Sequence showing Dijkstra's interpolation between landmark points

3.3.3 Single source multiple destinations shortest path algorithm using dynamic programming

Here we implement the Dijkstra'a algorithm for finding the shortest path between two nodes in an undirected weighted graph [27] . The pixels here are modeled as the nodes and the edge weights are the ones calculated in the previous section.

Dijkstra's Algorithm

Initialize a list with the seed pixel

while *the list is not empty* **do**
  Remove the minimum cost point from the list
  Flag the point as processed
  for *every neighbor of the current point that has not been processed* **do**
    Compute the total cost to the neighbor
    if *neighbor is in the list and the total cost is less than the current cost at the neighbor* **then**
      Remove the neighbor from the list
    end
    if *neighbor is not on the list* **then**
      Assign the total cost to the neighbor
      Set the pointer of the neighbor to point to the center pixel
      Push back the neighbor on the list
    end
  end
end

Algorithm 4: Dijkstra's shortest path algorithm

The Dijkstra's algorithm improves the detection accuracy in two ways. It helps offset the errors caused by incorrect ASM landmarks and provides a smoother fit to the ear contour as shown in Fig 3.15. Some sample results are provided in Fig 3.16

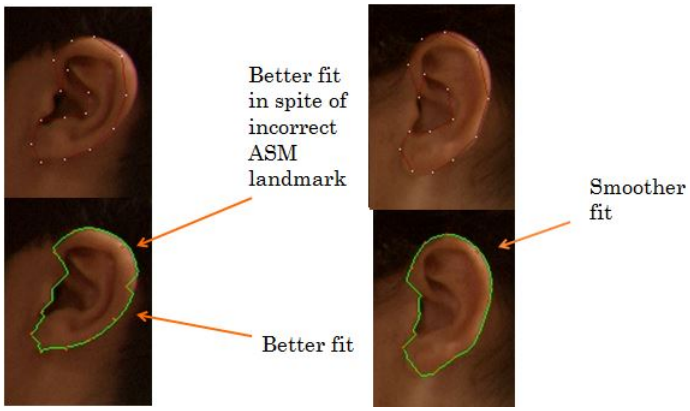

Figure 3.15: Improvement in output using Dijsktra's algorithm

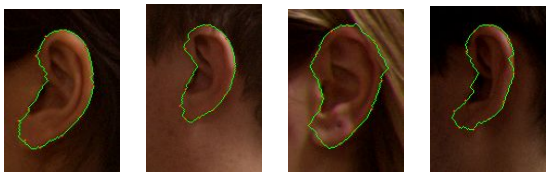

Figure 3.16: Sample outputs from Dijkstra's algorithm

3.3.4 Extracting the ear

Once the ear boundary has been detected, the next step is to extract the entire ear lying within the ear boundary. A number of point-in-polygon algorithms like ray casting, angle summation were considered but since the shape under consideration is a smooth curve and not a regular polygon, they were all computationally very expensive. Therefore, an inverted floodfill algorithm was used to find the set of points enclosed by the ear contour. A floodfill algorithm gives a set of connected components having the same color as the seed pixel. In this case, the only pixels that will not form a part of the connected components set will be the set of ear pixels since they are enclosed by the ear boundary. Thereby by selecting the set of all the pixels which are not common to the set of all image pixels and the set of connected component pixels, we can directly extract the ear pixels.

Inverted Floodfill Algorithm

Input: A binary image with a shape mask

Output: A binary image with the ear pixels highlighted

1. Input the image with the outer ear boundary.
2. Starting with pixel (0,0) as the seed pixel, initiate a floodfill algorithm
3. Invert the image.
4. Output binary image with the ear pixels highlighted.

Figure 3.17: Extracted ear using floodfill

In the next chapter, we will analyze and discuss some of the ear segmentation results that we obtained using the techniques described in this chapter.

Chapter 4

Results

The ear contour modeling and detection technique described in Chapter 3 was implemented using OpenCV 2.3, Blepo computer vision library, Microsoft Visual C++ and Matlab. It was tested on an Intel(R) Core(TM)2 Quad CPU 2.66 GHz with 8GB RAM. The detected ear contour was compared with manually annotated ear contours. The results are presented in this chapter.

4.1 Database

The database used for evaluation was the HID-J2 collection from the University of Notre Dame. The dataset we used had 413 subjects with varying ethnicity, gender, skin color and illumination. The average resolution of the original image was 640 X 480 and the resolution of the cropped ear image was between 94 X 132 to 125 X 182. The images we chose for our experiments were of the left side profile of the subjects.

4.2 Ground Truth

To get the ground truth data, we manually marked the contours around the ear using an interactive version of the Dijkstras algorithm. The number of landmark points varied from image to image depending on the image quality. For low quality images, more number of points were required for a better representation of the ear contour. The manual contour was marked on 331 images. Out of the 413 images, we used 82 images as the ASM training set. This consisted of 40 images selected

initially and 42 images in which the Viola Jones detector could not detect an ear.

4.3 Computing the Detection Rate

The next step is to devise a metric to detect how accurately the algorithm detected the ear shape. This section describes the detection criterion we used to classify a result as a true or false detection. Let, S_g be the set of all pixels in the ear mask in the ground truth data, S_d, the set of all ear pixels in the detected ear.

Then

$$S = S_d \cap S_g$$

where S is the set of common pixels between the ground truth and the detected ear.

$$D_s = \frac{cardinality(S)}{cardinality(S_g)}$$

$$E = S_d \not\cap S$$

$$D_e = \frac{cardinality(E)}{cardinality(S_d)}$$

D_s gives a measure of the accuracy of the detector. It is the fraction of the ground truth data that was correctly detected by the algorithm. D_e gives a measure of the error in the detected ear. It is the fraction of pixels that was present in the detected ear, but not in the ground truth. These two measures together give a good indication of the detection accuracy. Ideally we need a high value for S and a low value for E, which means for a perfect detection, $D_s = 1$,$D_e = 0$

4.4 Final Results

Viola Jones Haar Cascades correctly detected 331 ears out of 373 images with a detection rate of 88.7%. The ASM and Dijkstra's algorithm was only evaluated on these 331 images. On an average 90.62% of the total ear was captured by the ASM and Dijkstra's interpolation. The average error was 4.2%. In terms of the number of correct detections, depending on different thresholds we will get different detection rates. In about 93.35% of the images, more than 80% of the ear was detected. In 92.86% of the images the error was less than 10%. The results are summarized in the

subsequent figures. A graph of the error rate and detection rate according to the different thresholds in shown in Fig 4.3. Another graph showing the error rate and detection rate for every image is plotted in Fig 4.4

Figure 4.1: Images corresponding to downward peaks

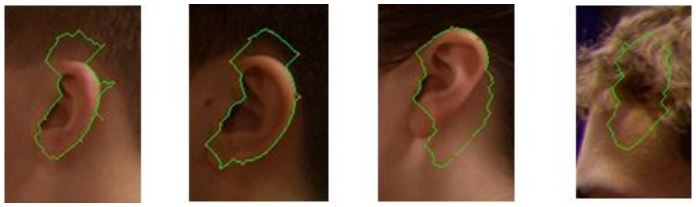

Figure 4.2: Images corresponding to upward peaks

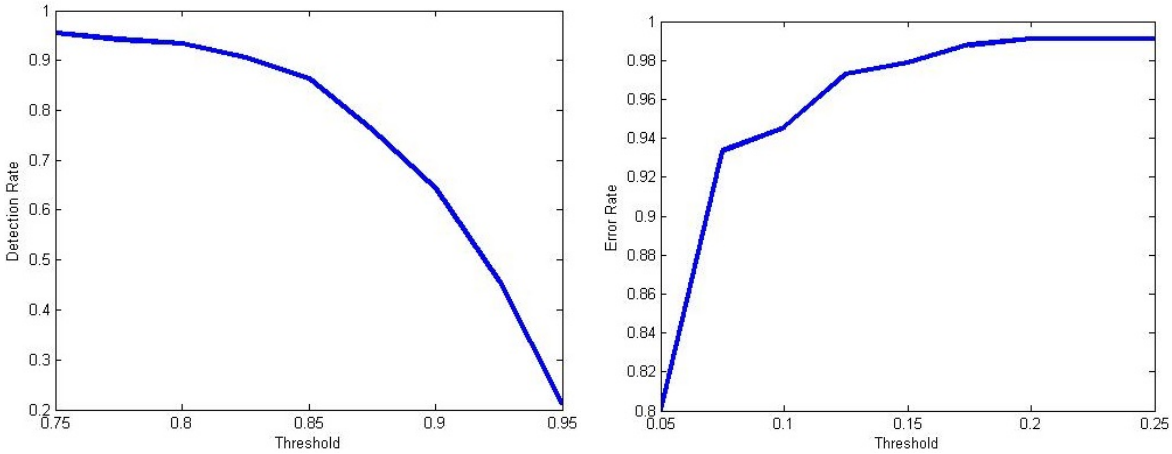

Figure 4.3: Detection and Error Rates

As can be seen, for the detection rate, if we consider the overlap threshold as 75% for true detection, then about 96.35% of the ears are correctly detected. As we increase the overlap threshold, the detection rate falls. Similarly for the error rate, if we keep the error threshold as 5%, then about

80% of the images have an error less than 5%. As we relax the error threshold, the detection rate increases. In Fig 4.4, the percentage overlap and the error for every image is plotted. The peaks correspond to the bad detections. The upward peaks indicate that the detected ear is much larger than the ground truth giving a large error and the downward peaks indicate that the detected ear is smaller than the ground truth giving a small error but a very low detection overlap. A good detection will have no upward or downward peak. Different thresholds for the detection and error percentages will give different detection rates.

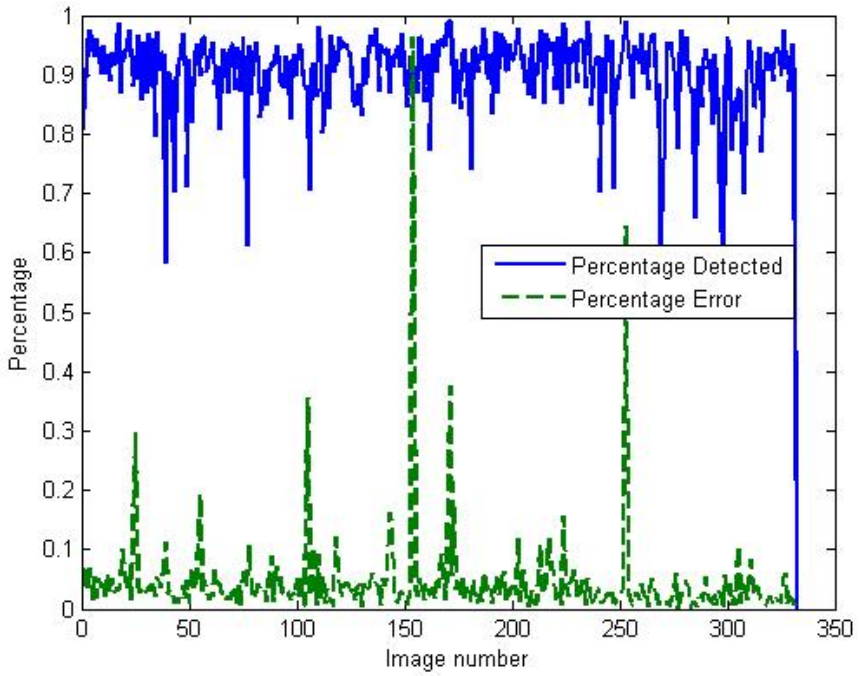

Figure 4.4: Detection and Error plot for all images

Some sample results for a low overlap is shown in Fig 4.1 and for a high error is shown in Fig 4.2.

4.4.1 Effects of illumination

Our algorithm was also tested on images with two kinds of illumination. One set of images consisted of images which were well illuminated and another set which was darker and lesser lit. The accuracy for both kinds of images was found to be comparable, with the lower illuminated

set performing only slightly worse than the bright images. The main reason for this robustness is that the Active Shape Model, which marks the key landmark points, is robust to variations in illumination. This is because the profile model is normalized to offset the effect of illumination. The results are summarized in Fig 4.5 to Fig 4.10.

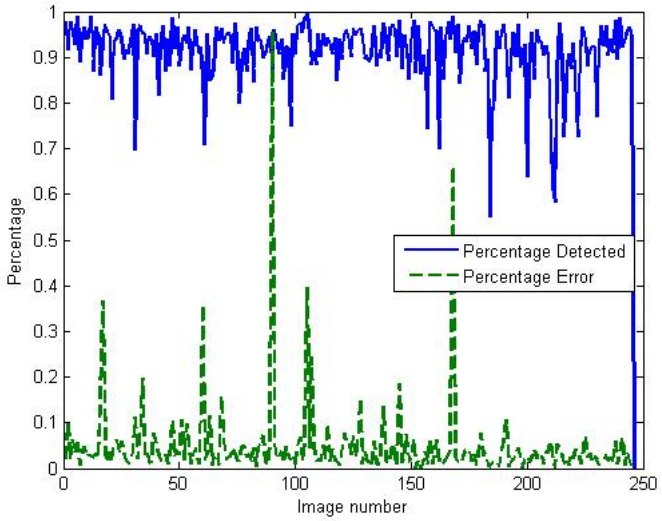

Figure 4.5: Detection and Error plot for bright images

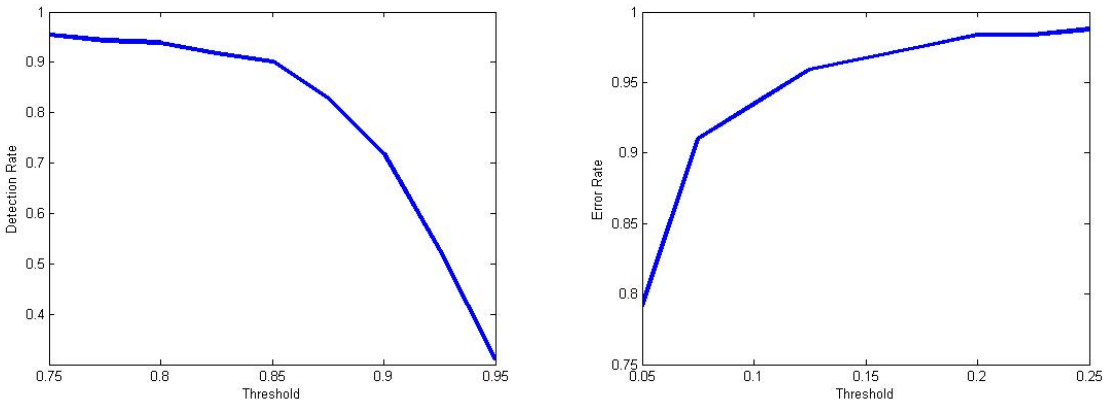

Figure 4.6: Detection and Error Rates for bright images

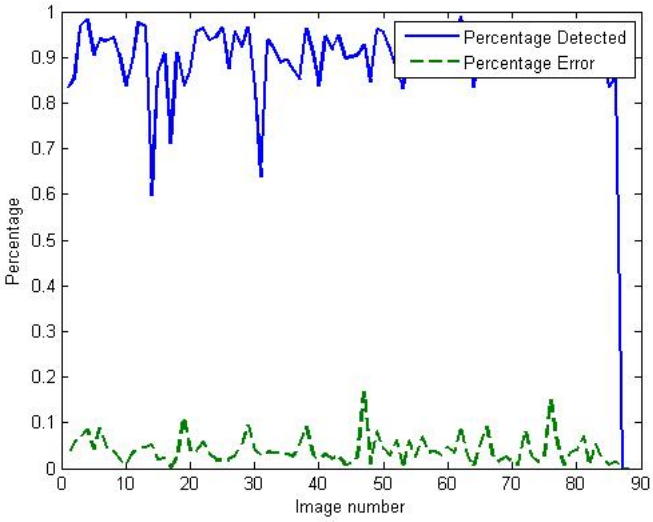

Figure 4.7: Detection and Error plot for dark images

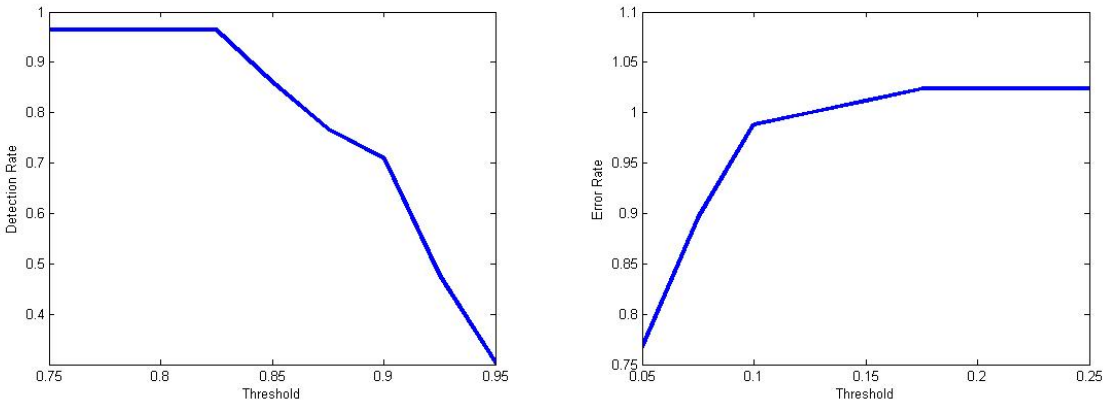

Figure 4.8: Detection and Error Rates for dark images

For the bright images, in Fig 4.8 it can be seen that the overlap accuracy is almost at 96% and stays stable until a threshold of 85%. The error rate is quite low too with around 92% having an error of less than 10%. As can be seen in Fig 4.8, the overlap accuracy is also quite high at around 98% for a threshold of 75-83% even for low lit images. The error is also quite low with over 95% having an error of less than 10%.

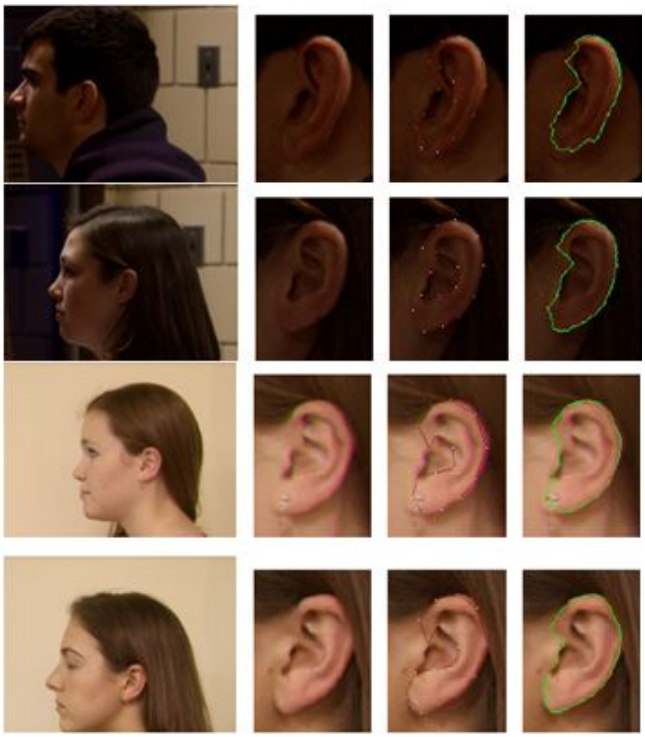

Figure 4.9: Sample results with different illumination

Illumination	Detection Rate (>80% overlap)	Error rate (<10%)	Mean error	Mean overlap
Bright/High (245)	93.88%	92.65%	4.54%	90.51%
Dark/Low (86)	96.51%	96.23%	3.95%	88.89%

Figure 4.10: Results summary for different illumination

4.4.2 Effect of skin color

The algorithm was also tested with subjects having different skin color tones. It was found that overall, the detection rate and accuracy was much higher for light skinned individuals compared to dark skinned individuals. This could be attributed to the fact that the Dijkstra's algorithm has a harder time detecting the strong contours in dark skinned subjects than light skinned subjects.

This is because of the low gradient variation in the ear region in dark skinned subjects. The results are summarized in Fig 4.11 to Fig 4.16

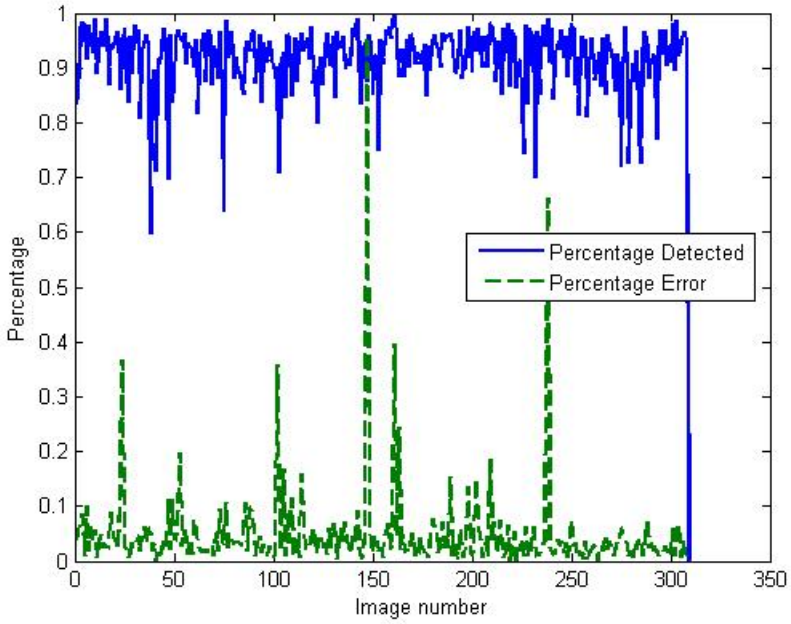

Figure 4.11: Detection and Error plot for light skinned subjects

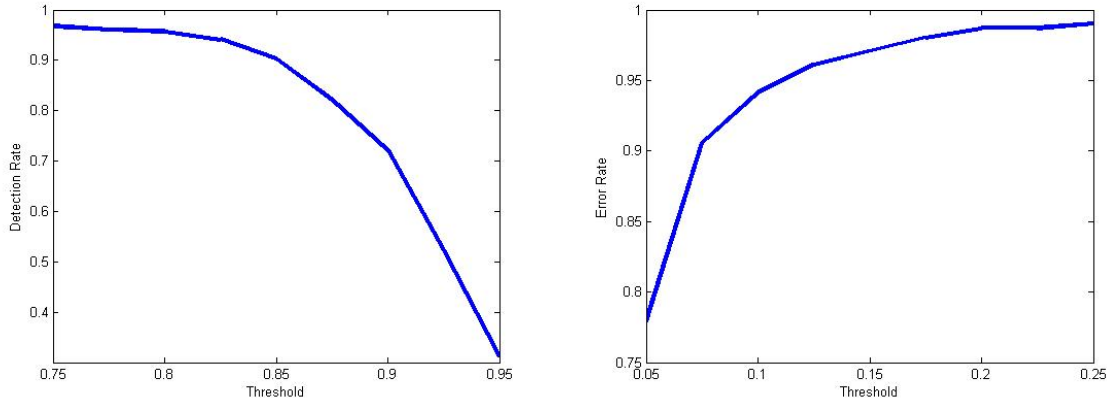

Figure 4.12: Detection and Error Rates for light skinned subjects

As can be observed from the detection rate plot in Fig 4.14, the overlap accuracy is quite low at around 81% with a threshold of 75%. The overall mean detection accuracy was only 79.24%

as compared to 90.96% for light skinned individuals. This is one of the areas where there is quite a lot of scope for improvement. The sample size for the dark skinned subjects was only 23. More experiments have to be performed on a larger database before devising any method to overcome this problem.

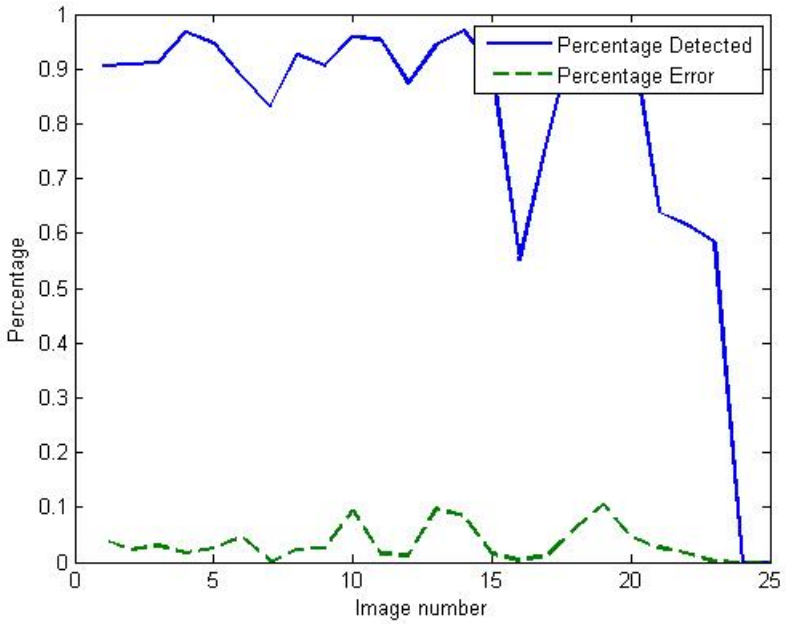

Figure 4.13: Detection and Error plot for dark skinned subjects

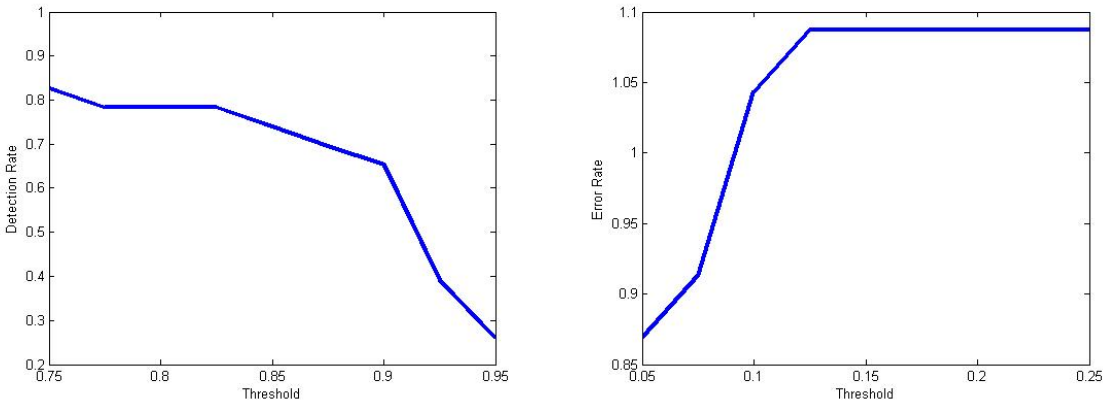

Figure 4.14: Detection and Error Rates for dark skinned subjects

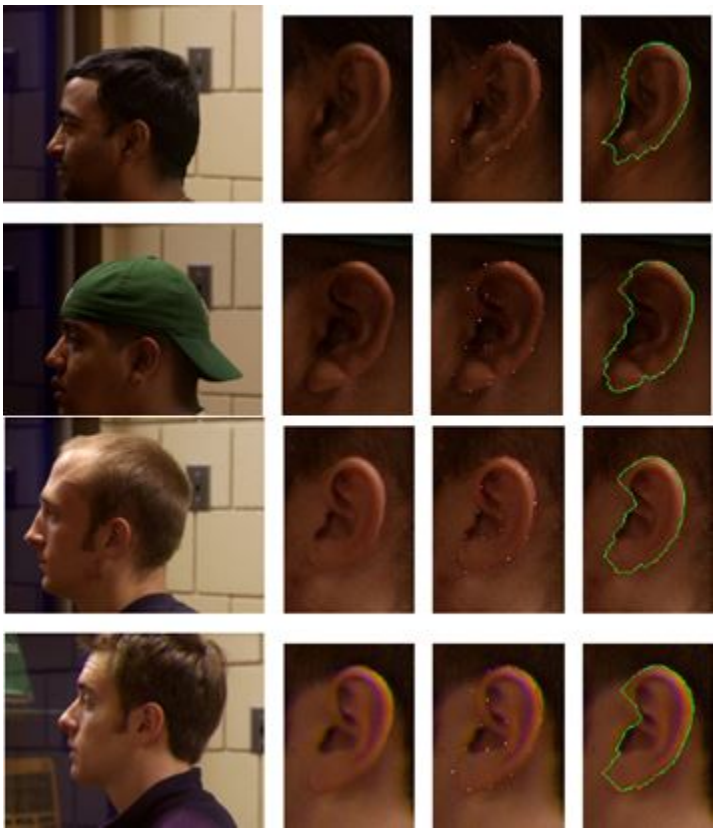

Figure 4.15: Sample results with different skin

Skin Color	Detection Rate (>80% overlap)	Error rate (<10%)	Mean error	Mean overlap
Light(308)	95.78%	93.51%	4.47%	90.95%
Dark (23)	78.26%	95.65%	3.31%	79.24%

Figure 4.16: Results summary for different skin

4.4.3 Computation time

The computational time for our algorithm was computed by the adding the times taken for the Viola Jones Haar Cascades detector, the Active Shape Model and the Dijkstra's algorithm.

$$T_{total} = T_{vj} + T_{asm} + T_{dijkstra}$$

Our algorithm was found to be much faster than other similar algorithms though it provides both a shape model for the ear and an accurate ear contour using only 2D color images. On an average it took about 1.77 seconds per image for the entire process. A table showing the time taken for some other ear detection methods is shown in the table below . Our approach also provides an accurate ear contour and a robust shape model for the ear, while the rest only provide one or the other. The image ray transform provides an ear contour but does not provide any ear model, the cascaded adaboost only gives a bounding box around the ear, the method using the histogram of categorized shapes requires 3D data and provides a robust shape model, but does not give mark the ear outer contour and the 3D model based approach provides a very weak geometrical model in terms of the 3D co-ordinates of the ear helix and anti-helix.

Method Used	**Time in seconds**	**Detection Rate**
Image Ray Transform [**8**]	5.45	99.6% (252 2D images)
Cascaded Adaboost using Matlab [**11**]	26.4	100% (203 2D images)
Histogram of Categorized Shapes [**31**]	2.8	100% (142 3D images)
3D Model Based [**5**]	6.5	92.6% (312 3D images)
Our Approach(with $\geq$ 75% overlap)	**1.77**	**96.35%** (331 2D images)

Table 4.1: Computation time per image

4.4.4 Sample results

Some sample outputs of the various stages of the algorithm with high detection rates are shown in Fig 4.17 to Fig 4.24. Some more sample examples with poor detection rates and accuracy are shown in Fig 4.25 and Fig 4.26.

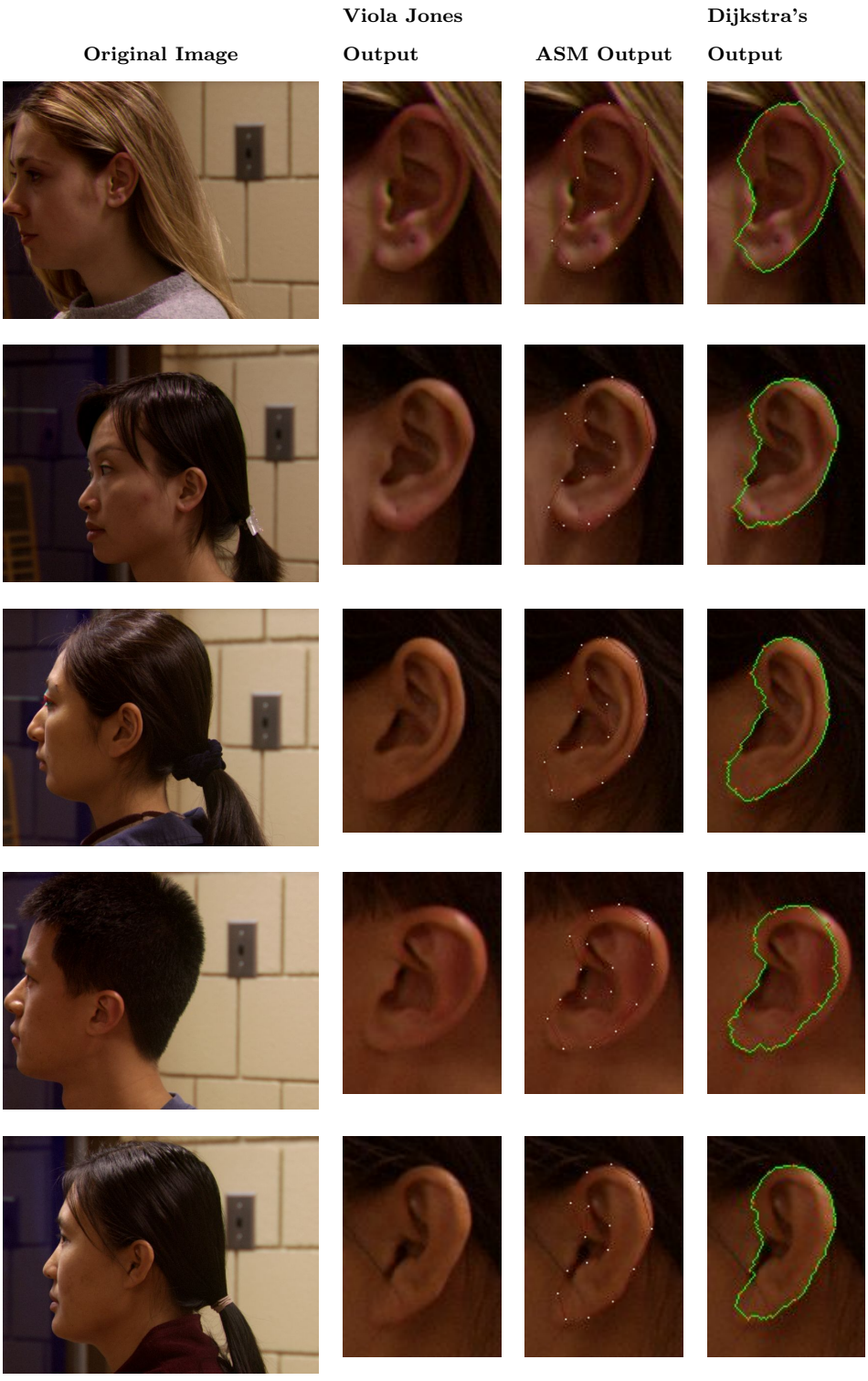

Figure 4.17: UND J2 Ear Segmentation Results - I [20]

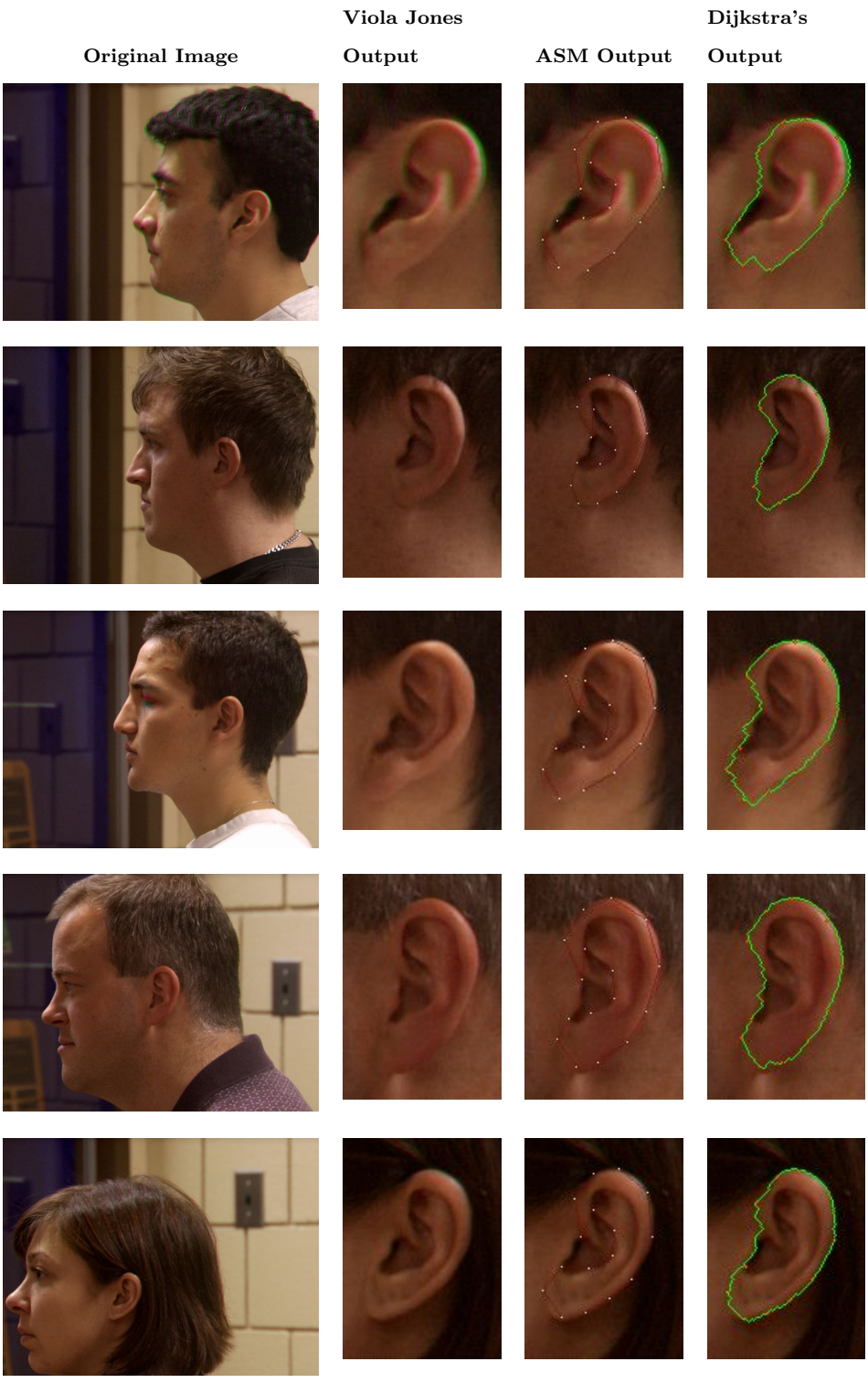

Figure 4.18: UND J2 Ear Segmentation Results - II [20]

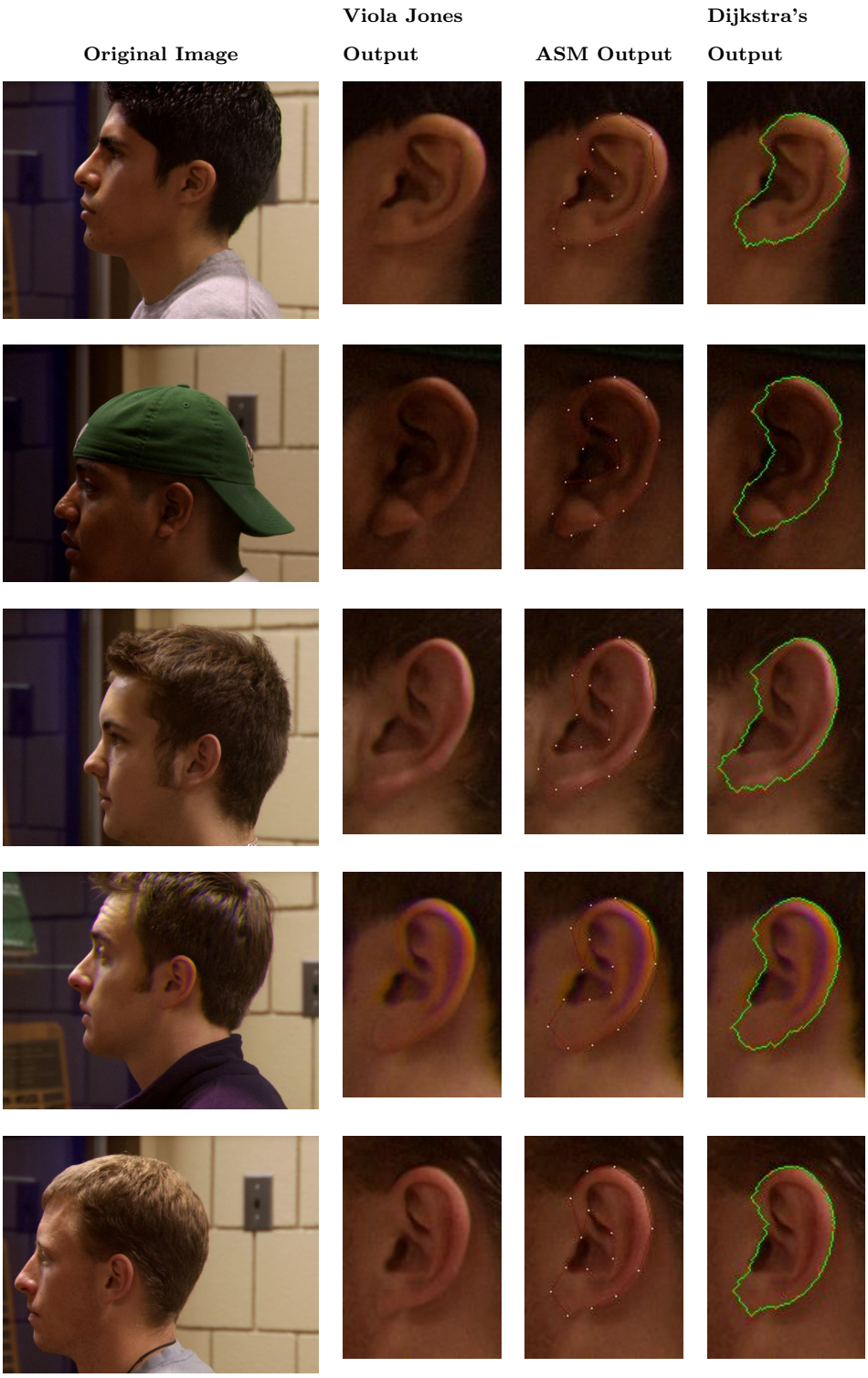

Figure 4.19: UND J2 Ear Segmentation Results- III [20]

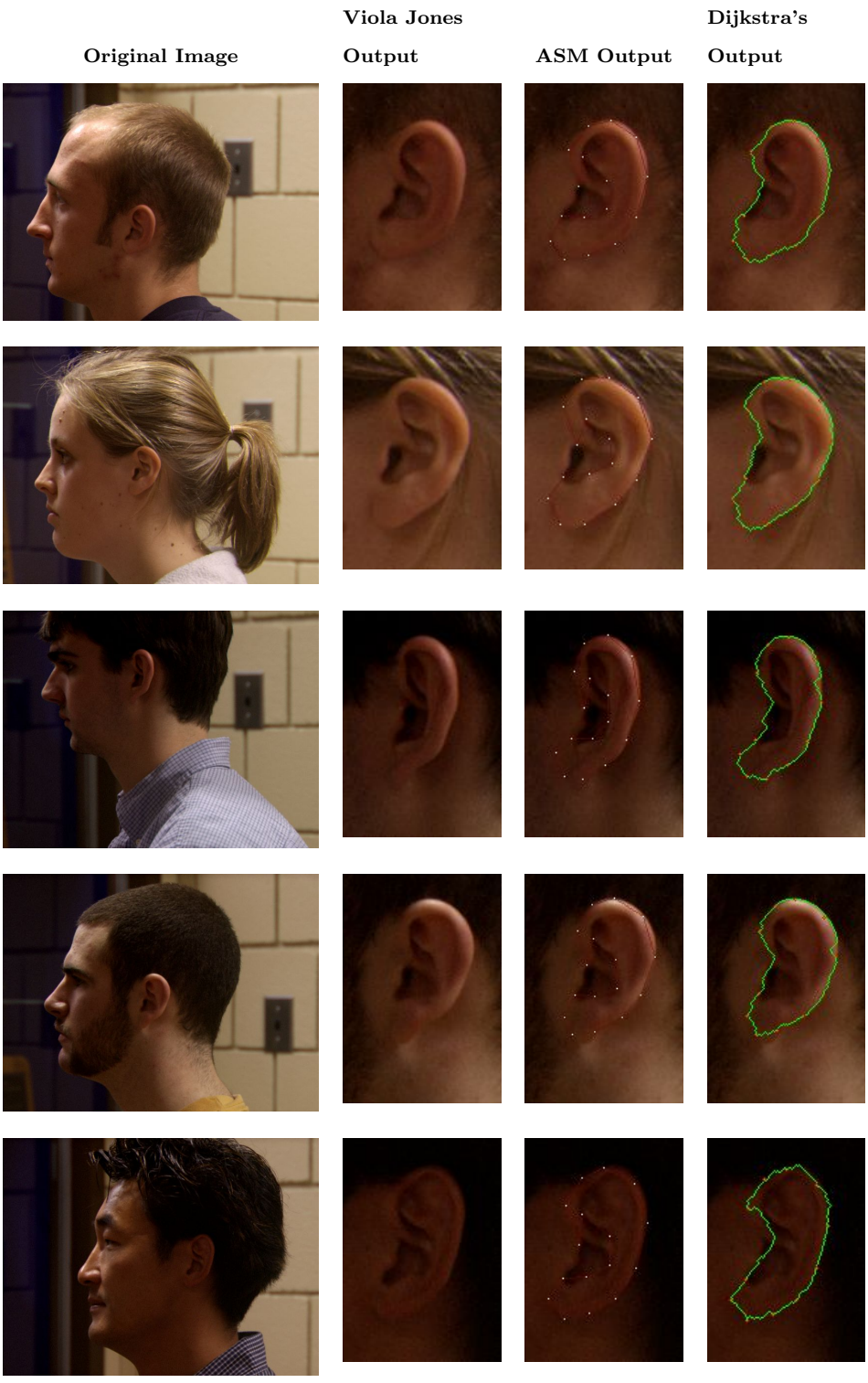

Figure 4.20: UND J2 Ear Segmentation Results- IV [20]

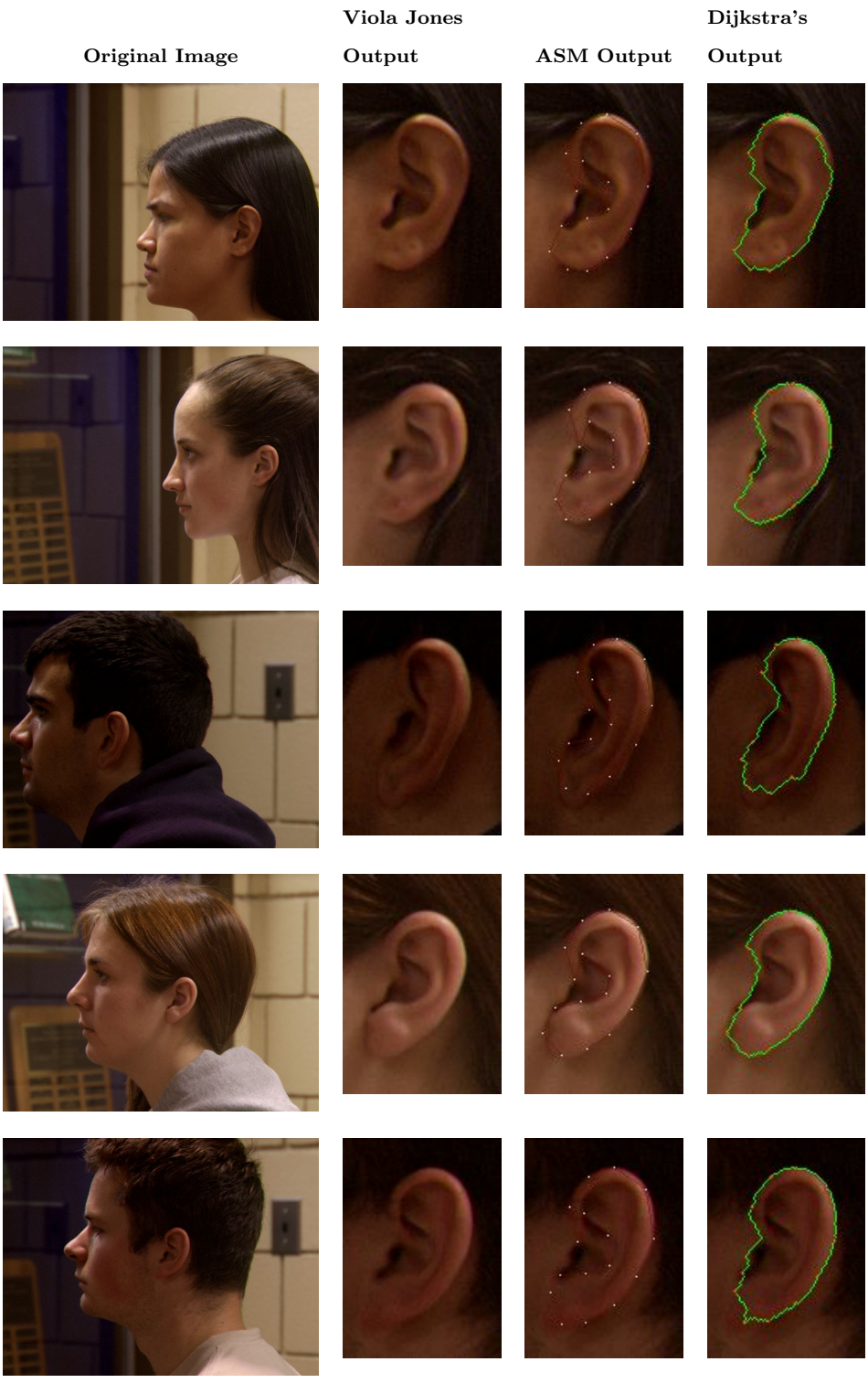

Figure 4.21: UND J2 Ear Segmentation Results- V [20]

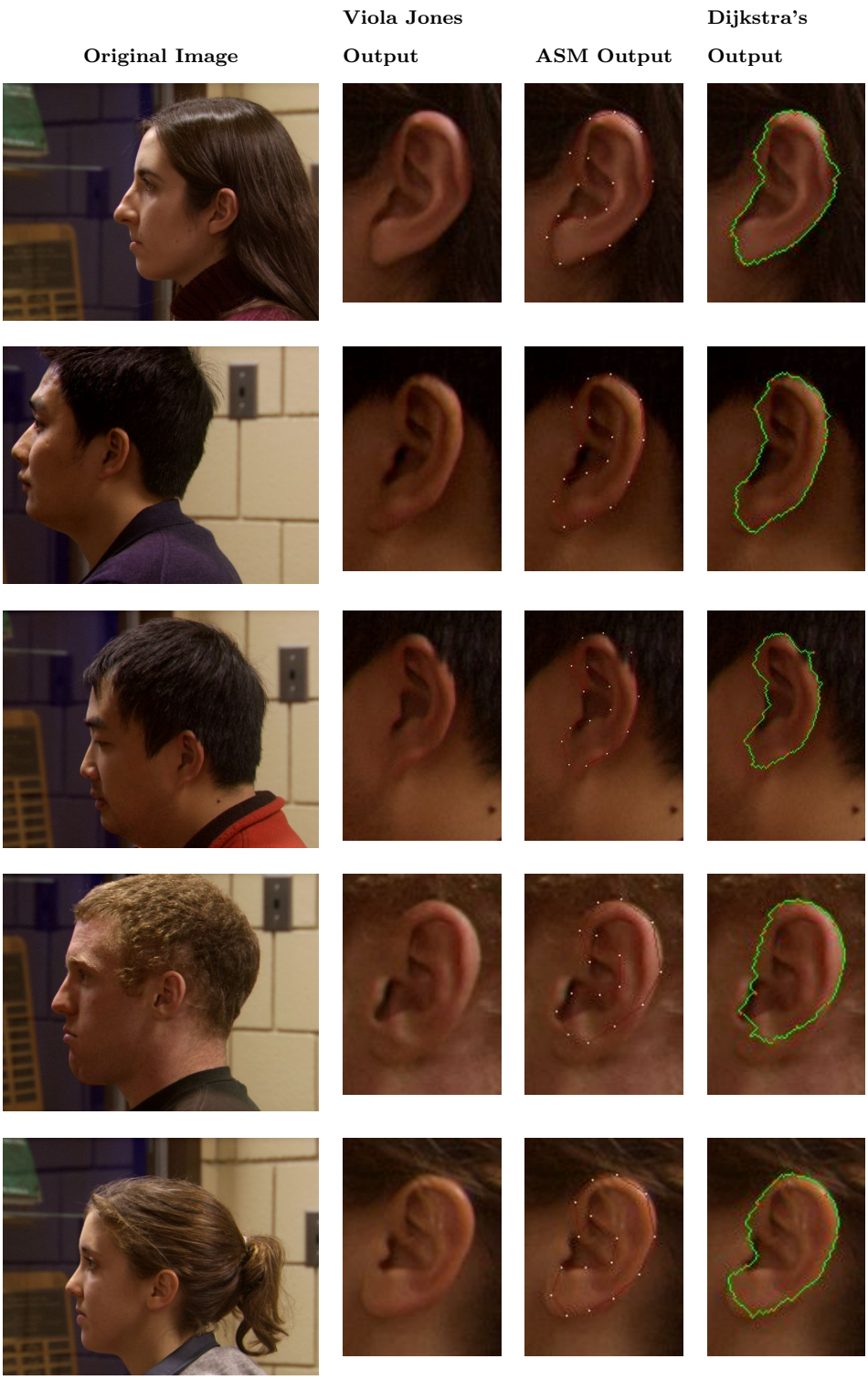

Figure 4.22: UND J2 Ear Segmentation Results- VI [20]

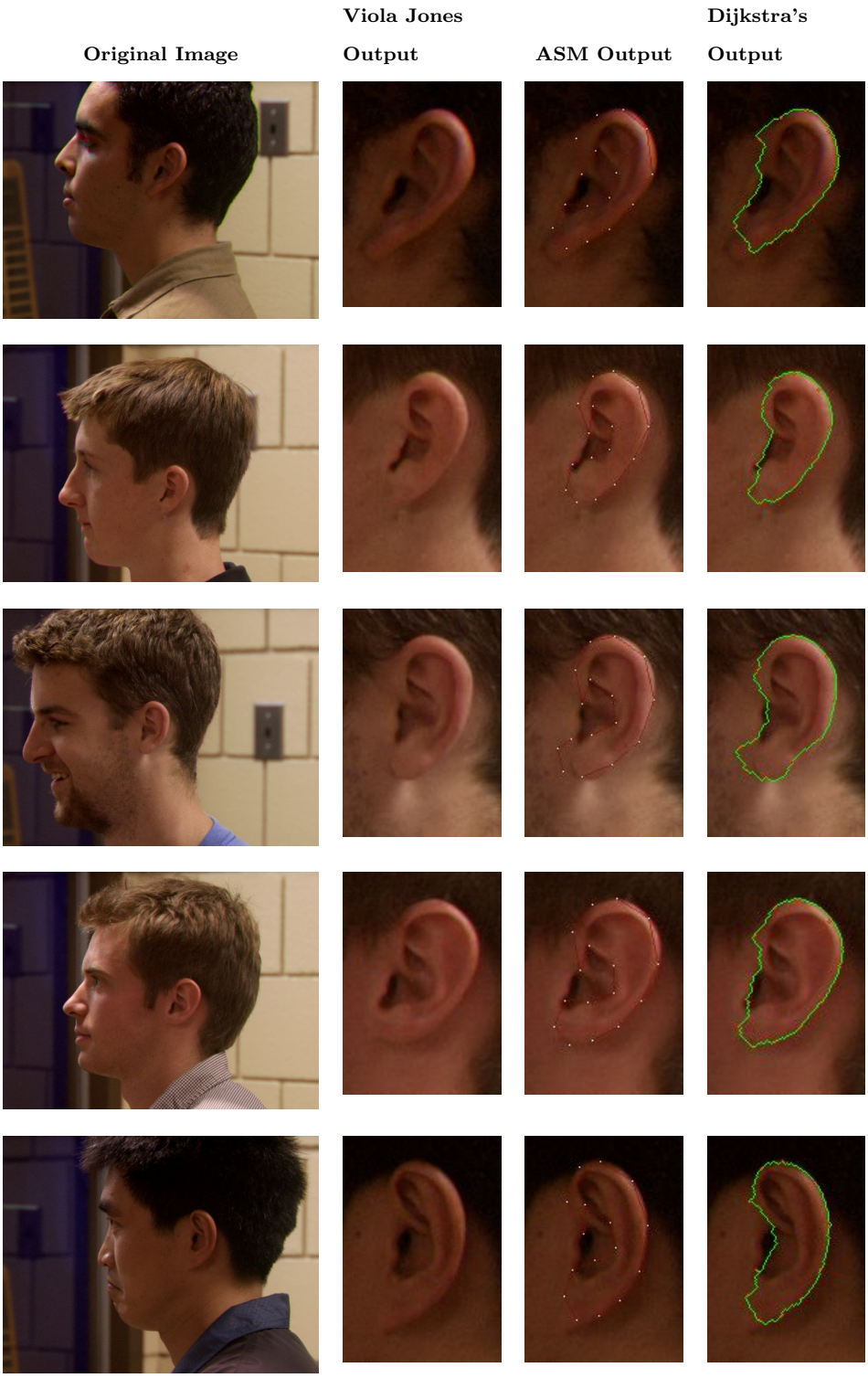

Figure 4.23: UND J2 Ear Segmentation Results- VII [20]

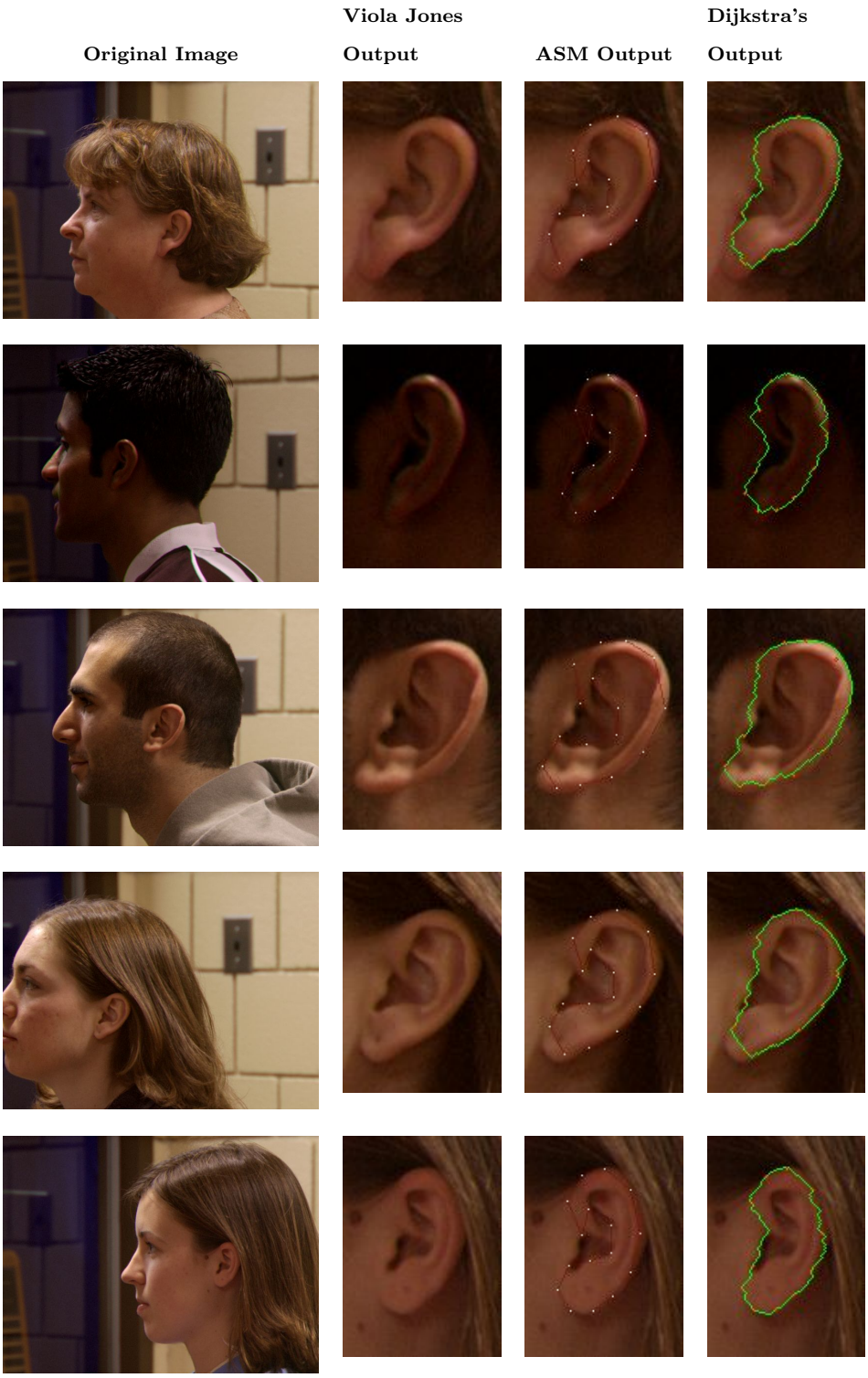

Figure 4.24: UND J2 Ear Segmentation Results- VIII [20]

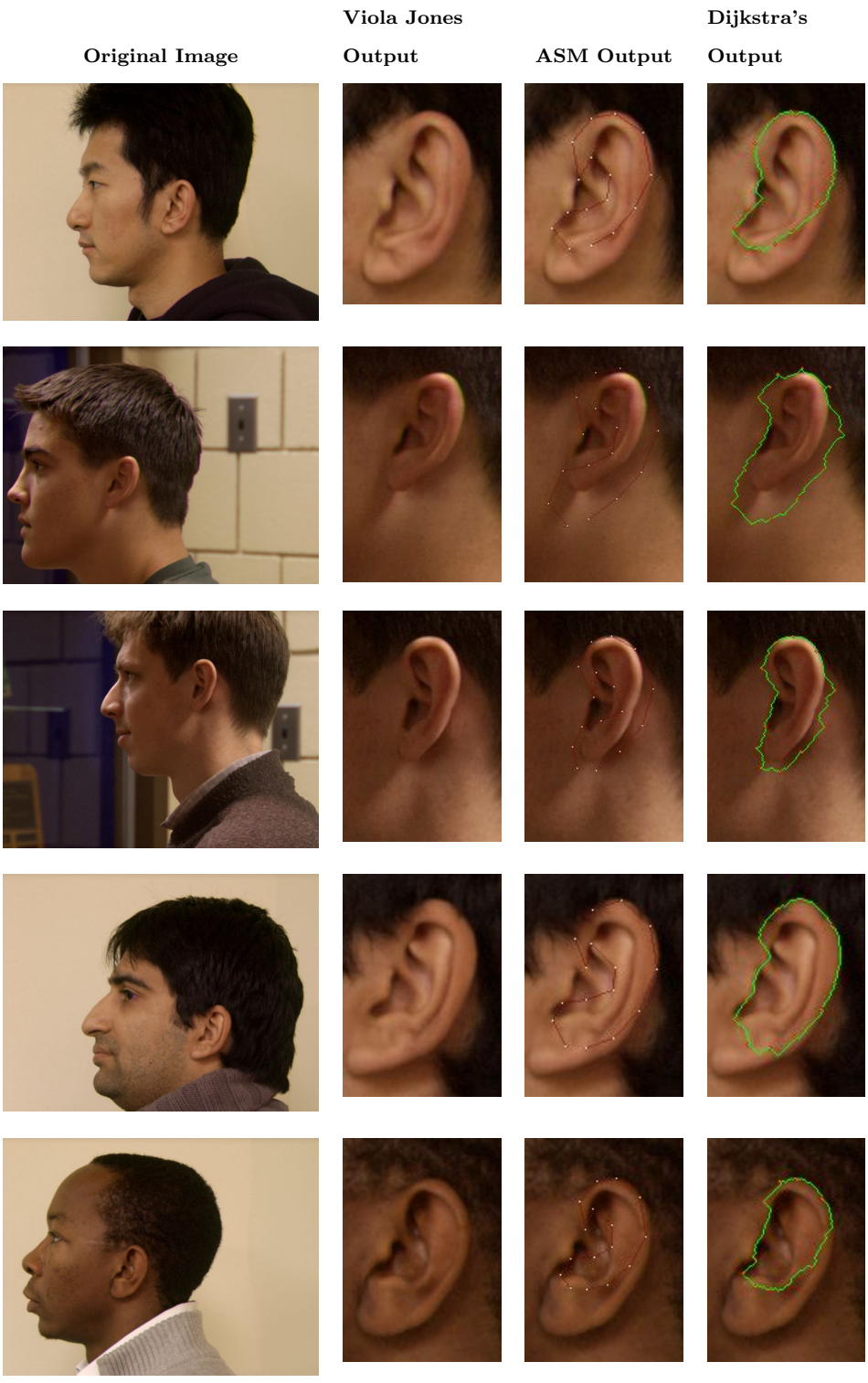

Figure 4.25: UND J2 Ear Segmentation Poor Results- I [20]

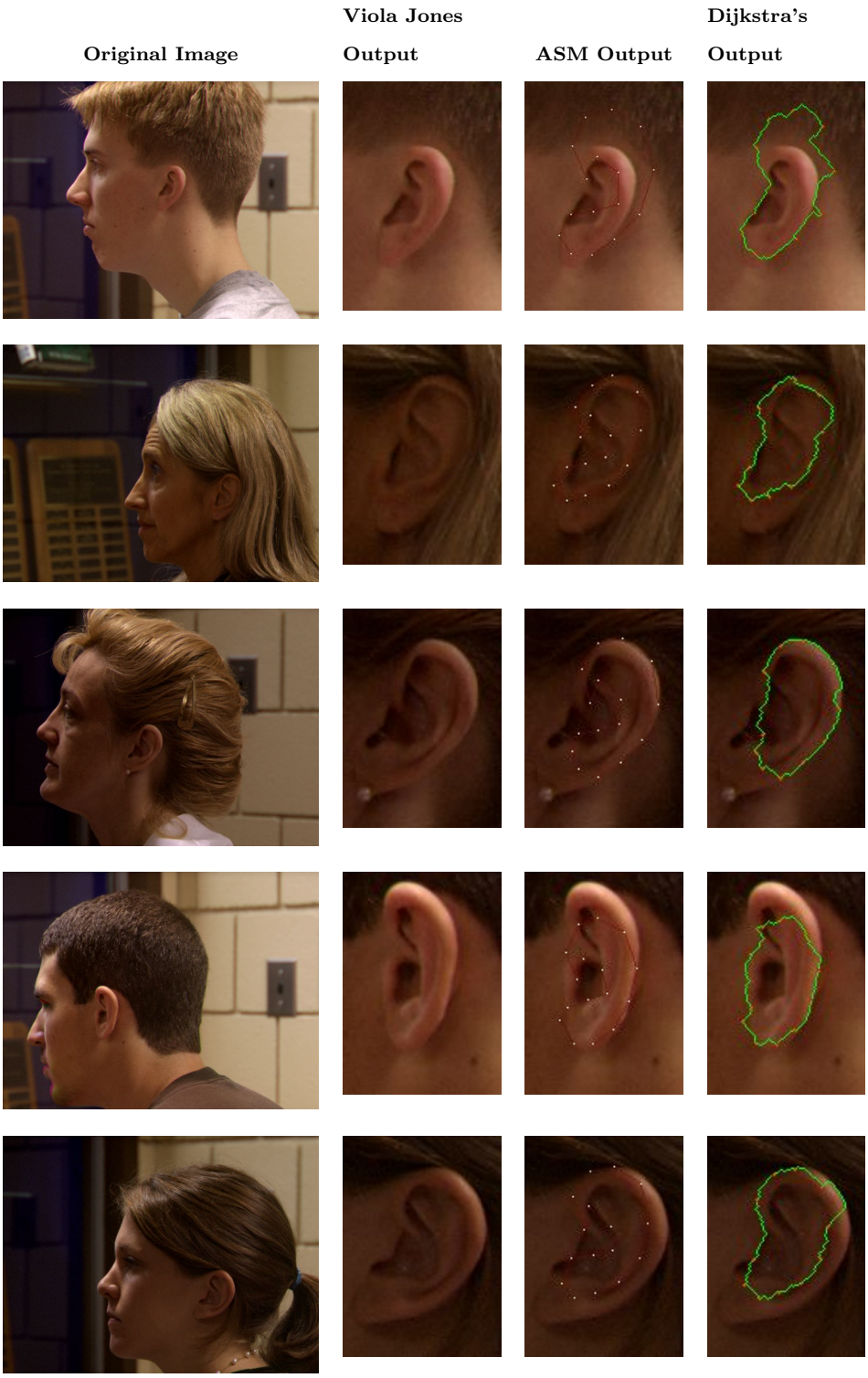

Figure 4.26: UND J2 Ear Segmentation Poor Results- II [20]

Chapter 5

Conclusions and Future Work

Ear detection is extremely important for various applications in Computer Vision and Biometric systems. In the tracking community, it has applications in head tracking and pose estimation. In the biometrics community, ear detection is an extremely important step in an ear recognition system. It has also been used to improve the accuracy of face detectors. Recent literature has focused on using the ear shape as a modality instead of the color and texture based features because of the unique and distinctive contours in the ear in both 2D and 3D data. In this thesis, we have proposed an end to end ear contour detection and modeling technique using a combination of Viola Jones Haar Cascades, Active Shape Models(ASM) and Dijkstras shortest path algorithm. The ASM provides a robust geometrical model which captures the variations in the ear shape using a set of parameters. Dijkstra's algorithm gives an accurate ear contour around the ear outer boundary. Apart from providing a shape based parametric model and an accurate ear outer contour, our algorithm has the additional advantage that the ear helix and the anti-helix can be extracted directly without any processing. This could be very important in part-based recognition schemes. It also gives a set of pre-established correspondences in the form of landmarks around the ear. In spite of providing a lot of information about the ear structure and shape, our algorithm still outperforms many similar algorithms in terms of computational efficiency while not compromising too much on the detection accuracy.

In the future, we hope to extend the work presented here to include the following features.

1. Use the ASM to perform biometric recognition. The ASM gives a set of basis vectors as the

mean shape model. Every ear shape is just a linear combination of these basis vectors. The set of coefficients for these basis vectors could be used as features for matching.

2. Use 3D active shape models to fit the ear in 3D space. Here, instead of profile models at the landmark points, we would have to compute surface models along the surface normal. It would also involve having more landmark points and a more complicated deformation model.

3. To use the 3D shape extracted here to perform recognition using 3D shape recognition techniques like the Iterative Closest Point [28]. We have already extracted the 3D shape from the 2D contour using the corresponding points in the HID-J2 3D database as shown in Fig 5.1.

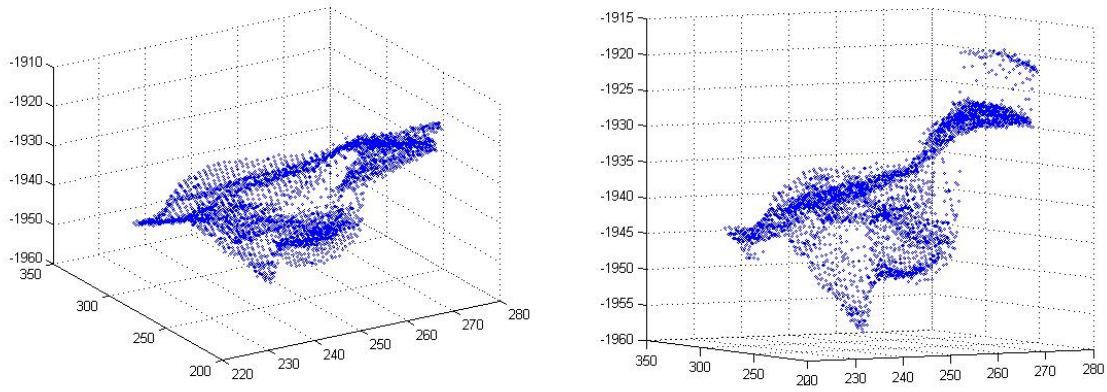

Figure 5.1: 3D ear shape extracted by our method

4. Use snakes and active contours instead of Dijkstra's algorithm to get the accurate ear contour [29]. This could be done by using the output of the ASM model as a starting point for the active contours, which can then deform and fit the ear outer contour.

Bibliography

[1] Ayman Abaza, Arun Ross, Christina Hebert, Mary Ann F. Harrison, and Mark S. Nixon. A survey on ear biometrics. *ACM Comput. Surv.*, 45(2):22, 2013.

[2] Saeeduddin Ansari and Phalguni Gupta. Localization of ear using outer helix curve of the ear. In *International Conference on Computing: Theory and Applications*, pages 688–692. IEEE, 2007.

[3] Banafshe Arbab-Zavar and Mark S Nixon. On shape-mediated enrolment in ear biometrics. In *Advances in visual computing*, pages 549–558. Springer, 2007.

[4] M. Castrillón Santana, J. Lorenzo Navarro, and D. Hernández Sosa. An study on ear detection and its applications to face detection. In *Conferencia de la Asociacin Espaola para la Inteligencia Artificial (CAEPIA)*, La Laguna, Spain, November 2011.

[5] Hui Chen and Bir Bhanu. Shape model-based 3d ear detection from side face range images. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 122–122. IEEE, 2005.

[6] Timothy Cootes. Statistical models of appearance for computer vision. University of Texas Research Report, 2004.

[7] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[8] Alastair H. Cummings, Mark S. Nixon, and John N. Carter. The image ray transform for structural feature detection. *Pattern Recognition Letters*, 32(15):2053–2060, November 2011.

[9] Kristen Grauman. Visual object recognition tutorial. University of Texas Lecture, 2012.

[10] Alfred V Iannarelli. *Ear identification*. Paramont Publishing Company California, 1989.

[11] Syed MS Islam, Mohammed Bennamoun, and Rowan Davies. Fast and fully automatic ear detection using cascaded adaboost. In *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*, pages 1–6. IEEE, 2008.

[12] Anil K Jain, Patrick Joseph Flynn, and Arun Abraham Ross. *Handbook of biometrics*. Springer, 2008.

[13] Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Trans. Circuits Syst. Video Techn.*, 14(1):4–20, 2004.

[14] Mohammad H Mahoor, Steven Cadavid, and Mohamed Abdel-Mottaleb. Multi-modal ear and face modeling and recognition. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 4137–4140. IEEE, 2009.

[15] Lynn Meijerman, Cor Van Der Lugt, and George JR Maat. Cross-sectional anthropometric study of the external ear. *Journal of forensic sciences*, 52(2):286–293, 2007.

[16] S. Milborrow and F. Nicolls. Active Shape Models with SIFT Descriptors and MARS. *VISAPP*, 2014. `http://www.milbo.users.sonic.net/stasm`.

[17] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *Computer Vision–ECCV 2008*, pages 504–513. Springer, 2008.

[18] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM, 1995.

[19] Erik Murphy-Chutorian and Mohan M Trivedi. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):607–626, 2009.

[20] University of Notre Dame. UND Profile Face Database, Collection J2, 2005.

[21] Anika Pflug and Christoph Busch. Ear biometrics: a survey of detection, feature extraction and recognition methods. *Biometrics, IET*, 1(2):114–129, 2012.

[22] Surya Prakash and Phalguni Gupta. An efficient technique for ear detection in 3d: Invariant to rotation and scale. In *Biometrics (ICB), 2012 5th IAPR International Conference on*, pages 97–102. IEEE, 2012.

[23] RAPP. Tutorial on integral images, 2011.

[24] Henry Taylor Fowkes Rhodes. *Alphonse Bertillon, father of scientific detection.* Abelard-Schuman, 1956.

[25] Chiarella Sforza, Gaia Grandi, Miriam Binelli, Davide G Tommasi, Riccardo Rosati, and Virgilio F Ferrario. Age-and sex-related changes in the normal human ear. *Forensic science international*, 187(1):110–e1, 2009.

[26] Teodora Vatahska, Maren Bennewitz, and Sven Behnke. Feature-based head pose estimation from images. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 330–335. IEEE, 2007.

[27] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[28] Ping Yan, Kevin W Bowyer, and Kyong J Chang. Icp-based approaches for 3d ear recognition. In *Defense and Security*, pages 282–291. International Society for Optics and Photonics, 2005.

[29] Ping Yan and K.W. Bowyer. Biometric recognition using 3d ear shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1297–1308, 2007.

[30] Li Yuan and Zhi-chun Mu. Ear recognition based on 2d images. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pages 1–5. IEEE, 2007.

[31] Jindan Zhou, Steven Cadavid, and Mohamed Abdel-Mottaleb. Histograms of categorized shapes for 3d ear detection. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–6. IEEE, 2010.