Chapter 2
Ear Detection in 2D

2.1 Introduction

Most of the well known ear biometric techniques have focussed on recognition on manually cropped ears and have not used automatic ear detection and segmentation. This is due to the fact that detection of ears from an arbitrary profile face image is a challenging problem as ear images may vary in scale and pose (due to in-plane and out-of-plane rotations) under various viewing conditions. However, for an efficient ear recognition system, it is desired to detect the ear from the profile face image in an automatic manner.

There exist few techniques in the literature which can be used to detect ear automatically. A detailed review of these techniques is as follows. The first well known technique for ear detection is due to Burge and Burger [1]. It has detected ears with the help of deformable contours. But contour initialization in this technique needs user interaction. As a result, ear localization is not fully automatic. Hurley et al. [2] have used force field technique to get the ear location. The technique claims that it does not require exact ear localization for ear recognition. However, it is only applicable when a small background is present in ear image. In [3], Yan and Bowyer have used manual technique based on two-line landmark to detect ear where one line is taken along the border between the ear and the face while other line is considered from the top of the ear to the bottom. The 2D ear localization technique proposed by Alvarez et al. [4] uses ovoid and active contour (snake) [5] models. Ear boundary is estimated by fitting the contour of an ear in the image by combining snake and ovoid models. This technique requires an initial approximated ear contour to execute and hence cannot be used in fully automated ear recognition system. There is no empirical evaluation of the technique.

Yan and Bowyer [6] have proposed another technique by considering a predefined sector from the nose tip as the probable ear region. It first computes the ear pit using the curvature information obtained from 3D data and uses its boundary to initialize active contour which detects the ear boundary. It fails if the ear pit is occluded. It produces 78.79 % correct ear segmentation when only color information is used for

S. Prakash and P. Gupta, *Ear Biometrics in 2D and 3D*,
Augmented Vision and Reality 10, DOI 10.1007/978-981-287-375-0_2

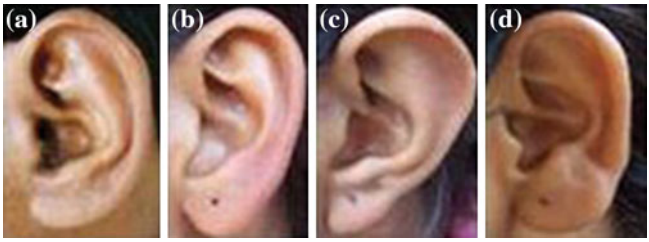

Fig. 2.1 Different ear shapes. **a** Round, **b** oval, **c** triangular, **d** rectangular

active contour conversion. Ansari and Gupta [7] have presented an ear detection technique based on edges of outer ear helices. The accuracy of the this technique is reported to be 93.34 % on 700 sample images collected at IIT Kanpur. The technique solely relies on the parallelism between the outer helix curves and does not use any structural information present in inner part of the ear and hence, it may fail if the helix edges are poor. Yuan and Mu [8] have proposed a technique based on skin-color and contour information. It detects ear by roughly estimating the ear location and by improving the localization using contour information. It considers ear shape elliptical and fits an ellipse to the edges to get the accurate position of the ear. There is no quantitative evaluation reported for the technique.

Another ear localization technique which exploits the elliptical shape of the ear has been proposed in [9]. It has been tested on 252 images of 63 individuals selected from XM2VTS [10] and 942 image pairs of 302 subjects of UND database. For XM2VTS database which is relatively small and has less complex images, the technique has achieved 100 % detection rate. However for UND database which contains complex images, it has offered only 91 % detection rate. Moreover, the assumption of considering ear shape elliptical for all subjects may not be true and hence, may not help in detecting the ear, in general. For example, as shown in Fig. 2.1, assumption of elliptical boundary may correctly approximate the ear boundaries for round and oval shapes but may fail in case of triangular and rectangular shapes. Also, this assumption restricts the ear localization to a controlled environment as the presence of background objects may produce false positives.

In [11], Sana et al. have given a template based ear detection technique where to detect ears at different scales, ear templates of different sizes are maintained. In practice, any predefined set of templates may not be able to handle all situations. Experimental study in this technique has used 1800 images collected from 600 individuals. However ear detection accuracy is not reported explicitly in the paper. In [12, 13], there are two techniques for ear localization which are also based on template matching. In these techniques, an ear template which is created off-line is resized to obtain a template of suitable size. Resizing is done using the size of the skin part of profile face image which works well when profile face includes only facial parts. But while capturing the profile face, an image may include other skin parts such as neck. This makes the size of the skin area larger than the actual and leads to an incorrect resizing of the ear template and hence, it produces an erroneous ear

localization. Techniques in [12, 13] have been tested on part of IIT Kanpur ear database containing profile face images of 150 individuals and found to have accuracy of 94 % and 95.2 % respectively.

Attarchi et al. [14] have proposed an ear detection technique based on the edge map. It relies on the hypothesis that the longest path in edge image is the outer boundary of the ear. It works well only when there is small background present around the ear and fails if ear detection is carried out in whole profile face image. Performance of ear detection of this technique has been reported on two databases, namely USTB database which contains 308 ear images from 77 persons [15] and Carreira-Perpinan database which includes 102 ear images from 17 persons [16]. Accuracy has been found to be 98.05 % for USTB database and 97.05 % for Carreira Perpinan database. A cascaded AdaBoost based ear detection approach has been proposed in [17]. The technique uses Haar-like rectangular features as the weak classifiers. AdaBoost is used to select good weak classifiers and then to combine them into strong classifiers. A cascade of classifiers is built which works as the final detector. The detection performance of the cascaded ear detector has been evaluated for 203 profile face images of UND database and is reported to have accuracy of 100 %. However, the technique needs huge amount of time for training and has been tested on relatively small set of images.

In [18], an ear localization technique has been proposed which is based on hierarchical clustering of the edges. To identify the edge cluster related to ear, the technique assumes approximate size of the ear cluster. Because of this, it works well when scale of the profile face image does not vary much. The technique is rotation invariant. However to handle scale, cluster size of the ear needs to be adjusted which may not be possible without user intervention. The technique has been tested on a database consisting of 500 profile face images of human profile faces collected at IIT Kanpur and found to have an accuracy of 94.6 %.

In [19], an ear detection technique using the image ray transform has been presented. The transform is capable of highlighting the tubular structures of the ear such as helix. The technique exploits the elliptical shape of the helix to perform the ear localization. However, assumption of ear shape being elliptical may be very rigid. The technique has achieved 99.6 % ear detection on 252 images of the XM2VTS database [10]. Ibrahim et al. [20] have employed a bank of curved and stretched Gabor wavelets (popularly called banana wavelets) for ear detection. A 100 % detection rate is achieved by this technique on images of XM2VTS database. In [21], a technique for ear detection has been presented by Kumar et al. where skin-segmentation and edge detection has been used for initial rough ear region localization. Region based active contour technique [22] has been further applied to get exact location of ear contours. The technique has been tested on 700 ear images and has achieved 94.29 % correct ear detection. This technique is applicable only when small background is present in the ear images. It can be observed that most of the techniques discussed above which have achieved almost 100 % correct ear detection rate have been tested on small data sets ($\leq$300 images).

Most of these techniques can detect the ear only when a profile face image contains a small background around the ear. These techniques are not very efficient,

particularly when profile face images are affected by scaling and rotation (pose variations). Moreover, they are not fully automatic and fast enough to be deployed in realtime applications. However, it is often required, specially in non-intrusive applications, to detect the ear from a whole profile face image which may be affected by scale and pose variations.

This chapter discusses an efficient ear localization technique which attempts to address these issues. The technique is invariant to scale, rotation and shape. It makes use of connected components of a graph constructed with the help of edge map of the profile face image to generate a set of probable ear candidates. True ear is detected by performing ear identification using a rotation, scale and shape invariant ear template.

Rest of the chapter is organized as follows. Section 2.2 briefly describes a skin color used for skin segmentation and Speeded Up Robust Features (SURF) used in ear template creation in the ear detection technique discussed in this chapter. Next section presents the ear detection technique. Rotation, scale and shape invariance of this technique has been discussed in Sect. 2.4. Experimental results are analyzed in Sect. 2.5.

2.2 Preliminaries

2.2.1 Color Based Skin Segmentation

This section presents a color based technique to segment skin and non-skin regions. It is similar to the skin segmentation technique proposed in [23] which has used 1976 CIE Lab color space for image representation. However, we have represented images in YCbCr space because it is perceptually uniform [24] and is widely used in video compression standards such as JPEG and MPEG [25].

The technique is capable of adapting different skin colors and lighting conditions. It performs skin segmentation in *YCbCr* color space as it is more suitable for characterizing skin colors. It first converts an image from *RGB* color space to *YCbCr* color space and then uses *YCbCr* color information for further processing. In *RGB* color space, *(R, G, B)* components represent not only color information but also luminance which may vary across a face due to the ambient lighting. This makes *(R, G, B)* components an unreliable measure for separating skin from non-skin regions. *YCbCr* color space separates luminance from the color information and hence, provides a way to use only color information for segmenting skin and non-skin regions.

The distribution of skin colors of different people is found to be clustered in a small area in the *YCbCr* color space. Although skin colors of different people may vary over a wide range, they differ more in brightness than its color. Due to this fact, skin color model is developed in *YCbCr* color space and only chrominance components (*Cb* and *Cr*) are used for modeling the skin pixels. Since color histogram of skin color distribution of different people is clustered at one place in *Cb*, *Cr* plane, it can be represented by a Gaussian model $N(\mu, \Sigma)$ with mean μ and covariance Σ.

With the Gaussian fitted skin color model, likelihood of skin for each pixel can be computed. If a pixel, having transformed from *RGB* color space to *YCbCr*, has a chromatic color vector $x = (Cb, Cr)^T$, the likelihood $P(x)$ of skin for this pixel can then be obtained by

$$P(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp[-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T] \tag{2.1}$$

Likelihood values obtained in Eq. 2.1 can be used to segment skin and non-skin regions. An adaptive thresholding process [23] is applied on likelihood image (obtained using skin likelihood values for all pixels) to compute an optimal threshold. Skin segmentation is obtained by thresholding the skin likelihood image using this threshold.

2.2.2 Speeded Up Robust Feature Transform

Speeded Up Robust Features (SURF) [26, 27] is a scale and rotation invariant interest point detector and descriptor. It has been designed for extracting highly distinctive and invariant feature points (also called interest points or key-points) from images. The reason behind using SURF for feature representation in this chapter (and also in Chaps. 3 and 5) is that it provides good distinctive features and at the same time is found to be more robust with respect to change in view point, rotation and scale, illumination changes and occlusion [27] as compared to other scale and rotation invariant shape descriptors such as SIFT [28] and GLOH [29].

There are two important steps involved in extracting SURF features from an image. These are finding key-points and computation of their respective descriptor vectors.

2.2.2.1 Key-Point Detection

SURF identifies salient feature points in the image called key-points. It makes use of hessian matrix for key-point detection. For a given point $P(x, y)$ in an image I, the hessian matrix $H(P, \sigma)$ at scale σ is defined as:

$$H(P,\sigma) = \begin{bmatrix} L_{xx}(P,\sigma) & L_{xy}(P,\sigma) \\ L_{yx}(P,\sigma) & L_{yy}(P,\sigma) \end{bmatrix}$$

where $L_{xx}(P, \sigma)$, $L_{xy}(P, \sigma)$, $L_{yx}(P, \sigma)$ and $L_{yy}(P, \sigma)$ are the convolution of the Gaussian second order derivatives $\frac{\partial^2}{\partial x^2}g(\sigma)$, $\frac{\partial^2}{\partial x \partial y}g(\sigma)$, $\frac{\partial^2}{\partial y \partial x}g(\sigma)$ and $\frac{\partial^2}{\partial y^2}g(\sigma)$ with the image I at point P respectively.

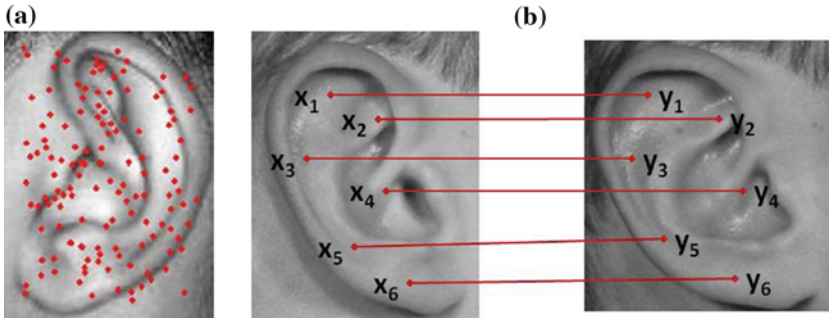

Fig. 2.2 Example of SURF features and matching, **a** SURF feature points, **b** matching

To speed up the computation, second order Gaussian derivatives in Hessian matrix are approximated using box filters. To detect key-points at different scales, scale space representation of the image is obtained by convolving it with the box filters. The scale space is analysed by up-scaling the filter size rather than iteratively reducing the image size. In order to localize interest points in the image and over scales, non-maximum suppression in a $3 \times 3 \times 3$ neighborhood is implemented. Figure 2.2a shows an example where SURF feature points.

2.2.2.2 Key-Point Descriptor Vector

In order to generate the descriptor vector of a key-point, a circular region is considered around each detected key-points and Haar wavelet responses dx and dy in horizontal and vertical directions are computed. These responses are used to obtain the dominant orientation in the circular region. Feature vectors are measured relative to the dominant orientation resulting the generated vectors invariant to image rotation. Also a square region around each key-point is considered and it is aligned along the dominant orientation. The square region is divided into 4×4 sub-regions and Haar wavelet responses are computed for each sub-region. Sum of the wavelet responses in horizontal and vertical directions for each sub-region are used as features. In addition, the absolute values of responses are summed to obtain the information about the polarity of the image intensity changes. Thus, the feature vector V_i for ith sub-region is given by

$$V_i = \{\Sigma dx, \Sigma dy, \Sigma |dx|, \Sigma |dy|\}$$

SURF descriptor vector of a key-point is obtained by concatenating feature vectors $V_i s$ from all sixteen sub-regions around the key-point resulting a descriptor vector of length $16 \times 4 = 64$. This is called SURF-64. Extended version of SURF (known

as SURF-128) which is more distinctive, adds a couple of more distinctive features to the descriptor vector. It uses the sums same as described above, however splits these values up further. It computes the sum of d_x and of $|d_x|$ separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sum of d_y and of $|d_y|$ are found according to the sign of d_x, hence doubling the number of features elements in the descriptor vector.

2.2.2.3 Matching in SURF

Matching in SURF is performed using nearest neighbor ratio matching. The best candidate match for a key-point of an image in another image is found by identifying its nearest neighbor in the key-points from the second image where nearest neighbor is defined as the key-point with minimum Euclidean distance from the given key-point of first image with respect to their descriptor vectors. The probability that a match is correct is determined by computing the ratio of distance from the closest neighbor to the distance of the second closest one. A match is declared successful if the distance ratio is less than of equal to a predetermined threshold $\tau \in (0, 1)$. Algorithm for SURF matching is described in Algorithm 2.1. Figure 2.2b shows an example where SURF matching points between two ear images are shown.

Algorithm 2.1 SURF Matching

- **Input:** Two sets of descriptor vectors $D^1 = \{D_1^1, D_2^1, \ldots, D_n^1\}$ and $D^2 = \{D_1^2, D_2^2, \ldots, D_m^2\}$ corresponding to n and m key-points of images I_1 and I_2 to be matched and matching threshold $\tau \in (0, 1)$.
- **Output:** Matching score N stating number of matching descriptor vectors in two images.

1: $N \leftarrow 0$
2: **for** $i = 1$ to n **do**
3: **for** $j = 1$ to m **do**
4: Compute $distance[j] = Euclidian\_Distance(D_i^1, D_j^2)$.
5: **end for**
6: Compute $[Sort\_Dist, Original\_Index] = Sort(distance)$ where $Sort(.)$ is a function which sorts *distance* array in ascending order and returns sorted distance values in array *Sort_Dist* and their corresponding original index values of *distance* array in array *Original_Index*.
7: **if** $\frac{Sort\_Dist[1]}{Sort\_Dist[2]} \leq \tau$ **then**
8: Descriptor D_i^1 of image I_1 matches to descriptor $D^2_{Original\_Index[1]}$ of image I_2 where *Original_Index*[1] is the index of the matched descriptor from image I_2.
9: $N \leftarrow N + 1$
10: **end if**
11: **end for**
12: Return matching score N.

2.3 Ear Detection Technique

This section presents an efficient ear detection technique. This technique is based on the fact that in a profile face image, ear is the only part which contains large variation in the pixel intensities, resulting this part rich in edges. This can be visualized from the image shown in Fig. 2.3f which displays the edge image of the skin segmented image of Fig. 2.3e. It can be observed that the ear part has larger edge density as compared to other parts. Further, it can also be noticed that all edges belonging to the ear part contain some curvature. These characteristics are exploited for ear localization in the presented technique which computes edge clusters in the edge map obtained from the profile face image and examines them for ear localization. Flow diagram of the technique is presented in Fig. 2.4.

2.3.1 Preprocessing

Profile face image undergoes a preprocessing phase before ear localization. This involves skin segmentation where skin areas of the image are segmented. Further, edge computation is carried out on skin segmented image. In the next step, obtained

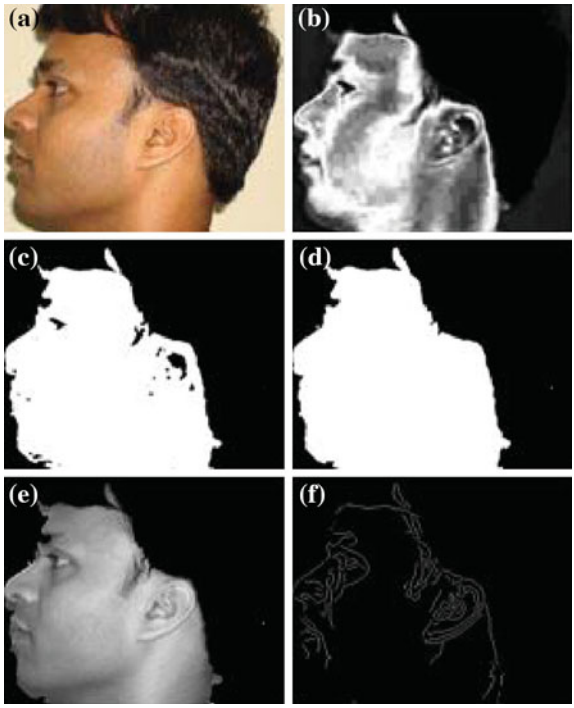

Fig. 2.3 Skin segmentation in profile face image. **a** Input color image, **b** skin-likelihood image, **c** binary image, **d** dilated binary image, **e** skin segmented image, **f** edge image

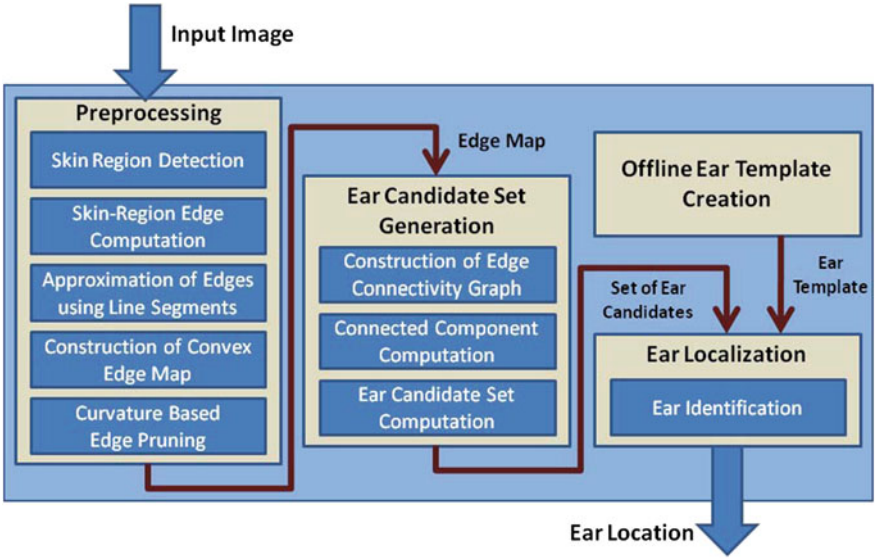

Fig. 2.4 Flow chart of 2D ear detection technique

edges are approximated using line segments and subsequently used in the construction of convex edge map. Erroneous edges are pruned out in the last step.

2.3.1.1 Skin Region Detection

Since ear exist in skin region, non-skin regions of the profile face should be segmented and removed from further processing. The skin color model discussed in Sect. 2.2.1 is used for skin segmentation. It transforms a color image into a gray scale image (called skin-likelihood image) using Eq. 2.1 such that the gray value at each pixel shows the likelihood of the pixel belonging to the skin. With an appropriate thresholding, the gray scale image is further transformed to a binary image segmenting skin (white pixels) and non-skin (black-pixels) regions. Since people with different skins have different likelihood, an adaptive thresholding process [23] is used to achieve the optimal threshold for each image.

The binary image showing skin and non-skin regions may contain some holes in it due to the presence of noise in the profile face image. Dilation is applied to fill these holes before using it for skin segmentation. The effect of this operation is to enlarge gradually the boundaries of regions of foreground pixels (i.e. white pixels). Thus the area of foreground pixels grows while filling holes within regions.

Figure 2.3 considers an example of skin region detection with various intermediate steps. For a color image given in Fig. 2.3a, corresponding skin-likelihood image is shown in Fig. 2.3b. It can be noticed that skin regions in Fig. 2.3b are brighter than the

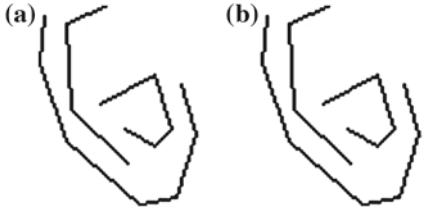

Fig. 2.5 Edge approximation: **a** original edge image, **b** edges approximated by line segments

non-skin regions. Figure 2.3c shows the binary image obtained by thresholding the skin-likelihood image. Dilation is applied on this image to repair it by filling small holes present in it. Figure 2.3d shows the repaired binary image. It is used for skin region detection where pixels of the profile face image corresponding to white pixels of the binary image are considered as skin pixels. Figure 2.3e shows the final skin segmented image. It can be observed from segmentation result that not all detected skin regions contain ear. Hence, ear localization can be used to locate the ear in all these skin like segments.

2.3.1.2 Skin-Region Edge Computation

Edge detection is carried out on skin segmented image using Canny edge operator and a list of all edges is computed. An edge in the list is obtained by connecting edge points together into a sequence of pixel coordinate pairs. Wherever an edge junction[1] is encountered, the sequence is terminated and a separate edge point sequence is generated for each of the branches and added to the list. This generates a set of edges containing two end points. Let χ be the set of all such edges.

2.3.1.3 Approximation of Edges Using Line Segments

All pixels present in an edge (belonging to set χ) may not be equally important and may not be necessarily required to represent the edge. So to remove redundant pixels from an edge and to get its compact representation, an edge is approximated using a set of line segments which keeps only those pixels which are important.

Line segments for an edge (belonging to set χ) can be found by considering array of edge points and finding the size and position of the maximum deviation from the line that joins the endpoints of the edge. If the maximum deviation exceeds the allowable tolerance, the edge is shortened to the point of maximum deviation and the process is repeated. In this manner each edge is broken into line segments, each of which adheres to the original data with a specified tolerance. Figure 2.5b shows an example of edge approximation by line segments for the edge image in Fig. 2.5a. Let χ_{ls} be the set containing all edges obtained after line segments fitting.

[1] Edge junction is a pixel where an edge divides into two or more edges.

2.3.1.4 Construction of Convex Edge Map

It is observed that edges belonging to the ear have convex[2] nature. However, because of the presence of noise such as hair near the ear, often false edges join true ear edges and make them non-convex. It may lead to an improper ear localization. This usually happens with the outer helix edges of the ear. To avoid this, the derived edges with set χ_{ls} are broken into a set of convex edges. Let χ_{convex} be the set of all convex edges. Identification of convex and non-convex edges and breaking the non-convex edges into convex can be done as follows.

Algorithm 2.2 Construction of Convex Edge Map

- **Input:** Set χ_{ls} of edges approximated with line segments.
- **Output:** Set χ_{convex} of convex edges.

1: Define a null set χ_{convex}.
2: **for** $\forall e \in \chi_{ls}$ **do**
3: Compute ρ_e using Eq. 2.2.
4: **if** $\rho_e == 0$ **then**
5: Add e to χ_{convex}.
6: **else**
7: Break e into a set of convex edges and add these edges to χ_{convex}.
8: **end if**
9: **end for**

Let there be an edge $e \in \chi_{ls}$ obtained after approximation. Assume e consists of k line segments with ith line segment, l_i, having end points: t_i and t_{i+1}. Let the line segment l_i be represented by vector $\overrightarrow{v}_i = t_{i+1} - t_i$. Let $\overrightarrow{v}_{i,i+1}$ be the vector cross-product of $\overrightarrow{v}_i$ and $\overrightarrow{v}_{i+1}$ (vector representing line segment l_{i+1}). The edge e is convex if directions of $\overrightarrow{v}_{i,i+1}$, for all i, are found to be same. To test whether an edge e is convex or non-convex, a decision parameter ρ_e can be estimated as follows.

$$\rho_e = \begin{cases} 0, \text{ if directions of vectors } \overrightarrow{v}_{(i,i+1)}, \forall i, \text{ are same} \\ 1, \text{ otherwise} \end{cases} \tag{2.2}$$

The edge e is convex if ρ_e is 0. To break a non-convex edge into a set of convex edges, it is scanned from one end to another and direction of each cross-product is analyzed. When a cross-product is found to be of different direction with respect to the previous cross-product, the edge is broken at that point. This procedure is continued till whole edge is broken into convex edges. Steps for construction of convex edge map are given in Algorithm 2.2.

Figure 2.6 presents an example of breaking of edges into convex type. Figure 2.6a shows two edges, one convex (edge *ABCD*) and another non-convex (edge *PQRS*). Vector representation of the line segments used in these edges and the direction of

[2] Edges which have curvature throughout either positive or negative are considered convex.

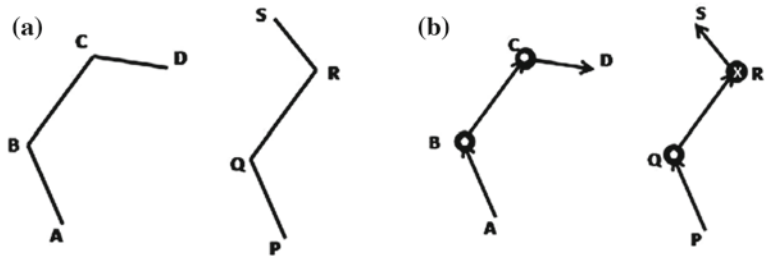

Fig. 2.6 An example of breaking edges into convex type: **a** edge map, **b** line segments with vector representation

the cross-products for adjacent vectors have been shown in Fig. 2.6b. Circle with a cross and circle with a dot at the joining points of two vectors represent the outward and the inward directions of the cross-product respectively. In edge *ABCD* of Figure 2.6b, it can be observed that all cross-products are inward so this edge is marked as convex while in edge *PQRS* of Fig. 2.6b, one cross-product is inward and other is outward so the edge is marked as non-convex. While scanning the edge *PQRS* from lower end, direction of the cross-product at point *R* is found to be different from the previous direction of the cross-product, so the edge *PQRS* is broken at point *R* into two edges: *PQR* and *RS*.

Ear localization accuracy can be improved by converting all non-convex edges to convex type. Breaking of non-convex edges into convex helps in removing the outlier edges (created due to noise). If the edges are converted to convex type, while constructing the edge connectivity graph, most of the outlier edges get isolated and do not appear in the connected component representing the ear and hence, do not affect the ear localization result. Figure 2.7 shows one such example of ear detection. In Fig. 2.7a, edge marked as *A* contains some erroneous part at its lower end arose due to the linking of true ear edge to a noisy edge present in the neck part. Due to this, when the edge *A* participates in the connected component representing ear, localization result includes some skin portion from the neck which does not belong to the ear. Figure 2.7c shows the localization result for this. When the edge *A* is segmented into convex edges *B* and *C* (Fig. 2.7b), lower part of the edge *A* (i.e. *B* after breaking) gets isolated from the ear edge cluster and remaining ear edge cluster produces the correct localization result. Figure 2.7d shows the localization result for this.

Any noise mainly affects the outer edge (helix) of the ear and hence, conversion of non-convex edges to convex primarily helps to remove noisy edges from the outer helix. Since detection of outer helix edge is difficult and computationally expensive, in the technique all edges are converted to convex type. However, conversion of non-convex edges present in the inner parts of the ear to convex type does not have any impact on the localization performance.

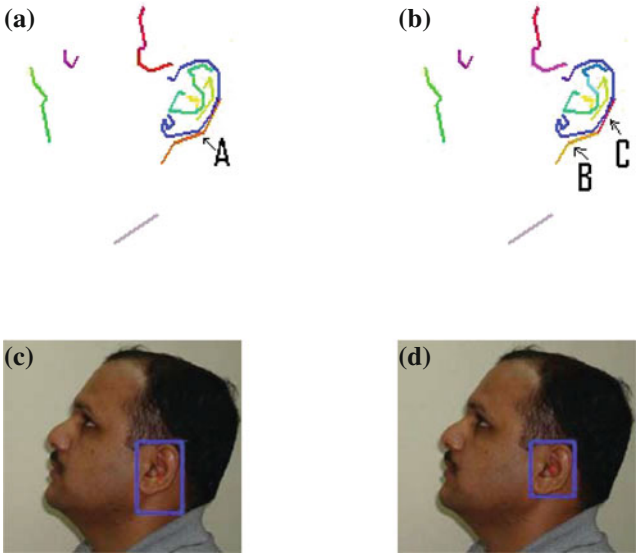

Fig. 2.7 Ear detection by breaking non-convex edges into convex edges where colors in (**a**) and (**b**) used to differentiate edges. **a** Edges Before convex type segmentation, **b** edges after convex type segmentation, **c** detected ear when (**a**) is used, **d** detected ear when (**b**) is used

2.3.1.5 Curvature Based Edge Pruning

All edges in the set χ_{convex} are of convex nature and are represented by line segments. It can be seen that each edge in the set χ_{convex} represented by one line segment (or two points) depicts a linear edge in the original edge map (set χ). Since all edges belonging to the ear contain some curvature, they need more than one line segment (or more than two points) for their representation. In other words, all edges having two points cannot be the part of ear edges and hence can be removed from the set χ_{convex}. This results a new edge set χ_c containing only the edges belonging to ear. Set χ_c can be formally defined as: $\chi_c = \{e \mid e \in \chi_{convex} \text{ and } \gamma(e) > 2\}$, where $\gamma(e)$ gives the number of points used in edge e to approximate it by line segments.

2.3.2 Ear Candidate Set Generation

This phase builds an edge connectivity graph which is used to find the connected components in the graph to obtain ear candidate set.

2.3.2.1 Building Edge Connectivity Graph

The set χ_c can be used to define the edge map of the profile face image. Let there be n edges in χ_c. The ith edge e_i in χ_c is defined by a point p_i. Thus χ_c can be represented by a set P of points $p_1, p_2, \ldots, p_n$ where p_i refers to e_i for all i. Against each edge e_i, a convex hull[3] $CH(e_i)$ is defined. If two convex hulls $CH(e_i)$ and $CH(e_j)$ intersect each other, then points p_i and p_j are connected through an arc[4] of a newly defined graph $G = (V, E)$ with the set of vertices V and the set of edges E, where

$$V = \{p_i \mid p_i \in P\}$$
$$E = \{(p_i, p_j) \mid CH(e_i) \text{ intersects } CH(e_j)\}$$

G is called edge connectivity graph. Algorithm 2.3 provides the steps invoked in building the graph G.

Algorithm 2.3 Construction of Edge Connectivity Graph

- **Input:** Edge map χ_c of profile face image I.
- **Output:** Edge connectivity graph $G = (V, E)$.

1: Define a graph $G = (V, E)$ where V and E are initially null.
2: Define a set $P = \{p_1, p_2, \ldots, p_n\}$ for the n edges in set χ_c such that point p_i represents ith edge e_i in set χ_c.
3: Define $V = \{p_i | p_i \in P\}$.
4: Define convex hull CH_i for each edge e_i, $e_i \in \chi_c$.
5: **for all** $i, j \in [1, n]$ **do**
6: **if** $CH(e_i)$ intersects $CH(e_j)$ **then**
7: Connect points p_i and p_j by an edge (p_i, p_j) in graph G and add it to E.
8: **end if**
9: **end for**
10: Return G.

One can observe that the ear edges are mostly convex in nature and if one moves from outer part of the ear towards inside, then most of the outer edges contain inner ear edges. Due to this nature of ear edges, convex hulls of the outer edges intersect the convex hulls of the inner edges. This almost guarantees that the convex hull of an edge belonging to the ear intersects at least another convex hull of the edge belonging to the ear. So this criterion to define connection between vertices (points) in a graph connects (directly or indirectly) all vertices belonging to the ear part with each other. Moreover, this criterion can define the connectivity irrespective of the scale; as a result, it makes the technique scale invariant. In general, property of one

[3] Convex hull for an edge is a tightest convex polygon which includes all edge points.

[4] In this work, "arc" signifies the notion of an edge in a graph. The word "edge" is used in the context of an edge in an image which is a set of connected pixels representing points of high intensity gradient in the image.

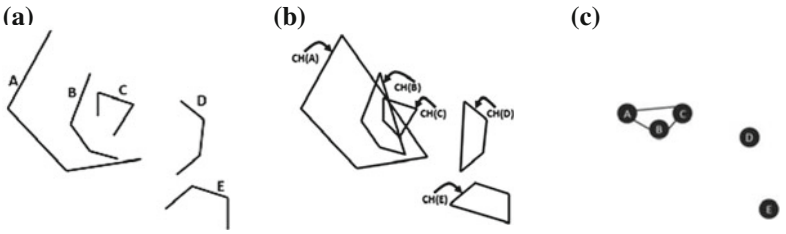

Fig. 2.8 An example of construction of edge connectivity graph. **a** Synthetic edges, **b** convex hulls, **c** connectivity graph

edge containing another is not true for the edges belonging to other parts of the profile face image; so vertices corresponding to these edges remain mostly isolated in the edge connectivity graph.

Figure 2.8 shows an example of an edge map and convex hulls of edges. It is seen from Fig. 2.8b that convex hulls of edges A, B and C intersect with each other. So vertices corresponding to these edges are connected to each other in the graph as shown in Fig. 2.8c. Points D and E are isolated in Fig. 2.8c since their respective convex hulls in Fig. 2.8b do not intersect to convex hull of any other edge.

It can be noted that there can be some simple criteria to define the connectivity among the vertices in edge connectivity graph. One such criterion may be based on some distance metrics between two edges. However, such choice makes ear detection scale dependent. This is due to the fact that the distance threshold required to define the connectivity among the vertices may vary for the images of different scales.

2.3.2.2 Connected Component Computation

Two vertices are in the same connected component of an undirected graph if there exists a path between them. After defining the graph for the edge map of profile face image, its connected components are computed. These components are analyzed one by one to localize the ear. To compute the connected components in the graph $G = (V, E)$, we have used a breath first search based algorithm described in [30].

Figure 2.9 presents an example of edge connectivity graph and connected components labeling. Figure 2.9a shows an edge image obtained from a profile face image. A graph, shown in Fig. 2.9b, is constructed for this edge image and connected components (enclosed inside rectangular boundaries) are computed. Magnified view of the component A present in Fig. 2.9b can be seen in Fig. 2.9c.

2.3.2.3 Ear Candidate Set Computation

Ideally, it is believed that the vertices representing ear edges are connected to each other (directly or indirectly) and form one connected component while all other vertices representing non-ear edges remain isolated. Hence the criterion based on

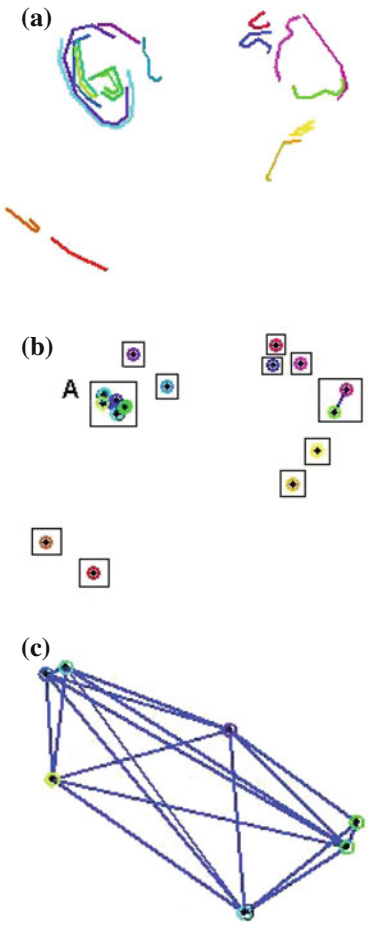

Fig. 2.9 An example of edge connectivity graph and connected components labeling. **a** Edge map (colors used to differentiate edges), **b** graph for the edge map of (**a**) with connected components labeling, **c** magnified view of component A of (**b**)

the size of the component can be used to find out the connected component representing ear. However, there may exist few more places in the profile face where due to noise a convex hull of one edge may intersect that of other edges and give rise to a large connected component. Hence, each connected component in the edge connectivity graph which has two or more vertices is considered as a probable candidate to represent the ear. Any connected component having single vertex can be straightaway removed from the graph as it cannot represent the ear. Let $K = \{K_1, K_2, \ldots, K_m\}$ be the set of connected components of graph G where each component has two or more number of vertices. Average vertex degree of a connected component K_j is defined as:

$$d(K_j) = \frac{\Sigma_{i=1}^{n_j} d(p_i)}{n_j} \tag{2.3}$$

where $d(p_i)$ is the degree of vertex p_i and n_j is the total number of vertices present in component K_j. As stated earlier, ear part of the profile face image is rich in edges due to large intensity variation in this region; hence, it is less probable that a connected component representing an ear will have only two vertices (or average vertex degree one). Therefore, to further prune out the false connected components, only the components having average vertex degree greater than one can be considered to obtain probable ear candidates. A probable ear candidate in a profile face image is defined as the image portion which is cropped using the bounding box of the edges participating in a connected component. A set of ear candidates is computed using all connected components satisfying the criterion on the average vertex degree. Algorithm 2.4 presents steps to generate ear candidate set using connected components.

Algorithm 2.4 Computation of Ear Candidate Set

- **Input:** Set $K = \{K_1, K_2, \ldots, K_m\}$ containing m connected components of G.
- **Output:** Set I_E containing the image portions cropped from the profile face which are the probable candidates for ear.

1: **for** $j = 1$ to m **do**
2: $d(K_j) = \frac{1}{n_j}\Sigma_{i=1}^{n_j} d(p_i)$, $p_i \in K_j$ and $d(p_i)$ is the degree of vertex p_i, n_j is the number of vertices in K_j.
3: **end for**
4: Define set $Q = \{j | d(K_j) > 1\}$
5: Define set $H = \{H_j | j \in Q\}$ where $H_j = \{e_i | p_i \in K_j, j \in Q\}$ contains edges with the edge e_i represented by point p_i in G as discussed in Sect. 2.3.2.1.
6: Define $B = \{B_j | B_j$ is the bounding box of the edges present in $H_j \in H\}$.
7: Obtain $I_E = \{I_j | I_j$ is cropped image from profile face using $B_j \in B\}$.
8: Return probable ear candidate set I_E.

2.3.3 Ear Localization

It is carried out by identifying the true ear among the probable ear candidates with the help of an ear template which is created off-line. The template works as an ear representative which depicts the characteristics of ears of various scales, rotations and shapes. Identification is performed by comparing the probable ear candidates with the ear template.

2.3.3.1 Ear Template Creation

To identify true ear, the template used for ear identification should exhibit the characteristics of scale and rotation invariance. To compute such a template in the presented

technique, a shape descriptor which is invariant to rotation and scale, is used. Among several scale and rotation invariant shape descriptors, SURF [27] provides good distinctive features and at the same time it is robust to changes in viewing condition, rotation and scale. Hence it has been used for ear template creation in this technique. As described in Sect. 2.2.2, SURF represents an image by first identifying some unique feature points in it and then by describing them with the help of a feature descriptor vector. For the description of the feature points, SURF uses intensity content within the neighborhood of feature point and describes it by using the sum of approximated 2D Haar wavelet components.

The ear template is computed by fusing the SURF feature descriptors obtained from various ear images together considering the redundant features only once. Let n be the number of ear images used for template creation. Let $T_1, T_2, \ldots, T_n$ be the SURF feature descriptor sets obtained from these images. A fused ear template T is obtained by

$$T = \bigcup_{i=1}^{n} T_i \tag{2.4}$$

If the set T_i contains c_i feature descriptor vectors, then total number of descriptor vectors c in T satisfies the following inequality

$$c \leq \sum_{i=1}^{n} c_i \tag{2.5}$$

Fusion of the SURF feature descriptor sets proceeds incrementally where first two sets T_1 and T_2 are fused to generate a new intermediate feature descriptor set which is further fused with feature descriptor set T_3. This process is continued till all sets are fused together. While fusing two SURF feature descriptor sets T_i and T_{i+1}, SURF matching (described in Algorithm 2.1) is performed between the two sets to find out the redundant feature descriptor vectors. If a descriptor vector in a set matches to a descriptor vector in another set, it is considered as common to both and is used only once in fusion. For example take the reference of Fig. 2.2b, if a feature point x_i from the first ear image matches to a feature point y_i in the second ear image, either descriptor vector for x_i or descriptor vector for y_i is used in fused feature descriptor set.

A SURF descriptor vector can be either of 64 dimensions or of 128 dimensions. A 128 dimensional descriptor vector provides more discriminative power as compared to 64 dimensional vector, however it involves more computation time. Since an ear template is used to discriminate between ear and non-ear candidates, experimentally it is found that it is sufficient to use 64 dimensional descriptor vector to create a good ear template.

It can be noted that attempts have been made to utilize the power of invariant feature points in other ear biometric systems as well. For example, Bustard and Nixon [31] have used Scale Invariant Feature Transform (SIFT) [28] feature points for registration of probe and gallery image before matching to perform ear recognition.

2.3.3.2 Ear Identification

Let the ear candidate set be $I_E = \{I_1, I_2, \ldots, I_\eta\}$ where η is the cardinality of set I_E and I_k is the image portion of the profile face image representing kth probable ear candidate, $k = 1, 2, \ldots, \eta$. For identification purpose, SURF feature descriptor set is computed for all the ear candidates in I_E. Identification of true ear is performed by comparing the ear template with the SURF descriptor sets of the ear candidates in I_E. Comparison between two SURF descriptor sets is performed using SURF matching which uses the ratio-matching scheme [28] to find out the number of descriptor vectors matching between the two sets. Let D_i and D_j be two descriptor vectors from sets S_1 and S_2 respectively. Let $d(D_i, D_j)$ be a distance metric between the descriptor vectors D_i and D_j. The descriptor vector D_i is said to be matched with D_j if

$$d(D_i, D_j) < \rho \times d(D_i, D_k), \;\; D_k \in S_2, k \neq j, \forall k \tag{2.6}$$

where ρ is a constant lying between 0 and 1. A small value of ρ gives a tighter matching while a large value of ρ provides a relaxed matching.

Let $T_E = \{T_{I_1}, T_{I_2}, \ldots, T_{I_\eta}\}$ be the SURF feature descriptor sets for the ear candidate images in I_E. To obtain the true ear, SURF matching is performed between ear template (T) and all elements of T_E and a match score vector *MatchScore* is generated. SURF matching between two descriptor sets returns the number of matched points between them. The true ear candidate I_ξ is obtained such that

$$\xi = \arg\max_i \{MatchScore[i]\}$$

That means, the ear candidate from I_E for which SURF match score is maximum, is declared as the true ear candidate. Algorithm 2.5 provides steps involved in ear identification process.

Algorithm 2.5 Ear Identification using SURF Descriptive Ear Template

- **Input:** Set $I_E = \{I_1, I_2, \ldots, I_\eta\}$ containing η probable ear candidates and off-line created ear template T.
- **Output:** I_ξ which is the true ear.

1: Define set $T_E = \{T_{I_1}, T_{I_2}, \ldots T_{I_\eta}\}$ where T_{I_i} represents SURF feature descriptor set for ear candidate $I_i \in I_E$.
2: **for** $i = 1$ to η **do**
3: $\quad MatchScore[i] = \text{SURFmatch}(T, T_{I_i})$.
4: **end for**
5: $\xi = \arg\max_i \{MatchScore[i]\}$.
6: Return I_ξ.

2.4 Scale, Rotation and Shape Invariance

In the technique discussed here, there are two major steps which play key role in ear localization. First step is the construction of edge connectivity graph which is used to detect probable ear candidates while second one is the identification of true ear among probable ear candidates using ear template. Construction of edge connectivity graph is made scale invariant by defining the connectivity among the vertices in the graph using intersection of convex hulls of corresponding edges. Such criterion to define the connectivity is unaffected by scale changes. Also, intersection of two convex hulls is unaffected if both are rotated; hence rotation also does not influence the process of defining the connectivity of two vertices in the graph. It can be observed that there is no significance of shape invariance at this step.

Rotation, scale and shape invariance at ear identification step is obtained by defining an ear template which exhibits these properties. It is achieved by using SURF feature descriptor for ear template creation which provides rotation and scale invariant description of ear feature points. An ear template is defined as a collection of rotation and scale invariant descriptor vectors obtained from multiple training ear images. Shape invariance is achieved by choosing the ears of different shapes from the database to define the ear template.

2.5 Experimental Results

The technique discussed here has been tested on three databases, namely IIT Kanpur (IITK) database and University of Notre Dame database—Collections E and J2 [32].

2.5.1 Estimation of Parameters

Parameters used for skin segmentation are computed for each data set separately by collecting few skin samples from each of them. Table 2.1 summarizes these parameters for various data sets. Minimum and maximum thresholds used in Canny edge detector are 0.0 and 0.1 respectively while standard deviation of the Gaussian filter σ is set to 1.0 for IITK database and 2.0 for UND database. Value of σ is kept little high for UND database as images in it are noisy. Distance tolerance for edge approximation is set to 20 for both the databases.

Ear template for each data set of IITK and UND databases has been created separately as the nature of the data present in each of them is entirely different. Few images are randomly selected from each data set to compute ear templates. It is found that 50 images from a data set are sufficient to capture the properties of the ears for creating a good ear template. The ratio value ρ used in SURF matching for template creation is taken as 0.5 whereas for true ear identification, it is set to 0.7. Since for

Table 2.1 Gaussian parameters used for skin segmentation in IITK and UND databases

Data set	Mean (Cb,Cr)	Covariance (Cb,Cr)
IITK data sets 1, 2 and 3	$\begin{pmatrix} 102.35 \\ 154.52 \end{pmatrix}$	$\begin{pmatrix} 71.76 & 9.95 \\ 9.95 & 111.77 \end{pmatrix}$
UND-E data set	$\begin{pmatrix} 90.65 \\ 170.23 \end{pmatrix}$	$\begin{pmatrix} 55.55 & -4.79 \\ -4.79 & 107.19 \end{pmatrix}$
UND-J2 data set	$\begin{pmatrix} 109.48 \\ 148.31 \end{pmatrix}$	$\begin{pmatrix} 55.74 & 41.76 \\ 41.76 & 93.62 \end{pmatrix}$

template creation, SURF matching is performed between the ear images, a lower value of ρ (which gives tighter matching) helps in capturing the distinct features of the ears. Ear identification is used to discriminate ear and non-ear candidates and hence matching is relaxed and little higher value of ρ is used.

2.5.2 Results

Figure 2.10 provides the results obtained at various steps of the ear detection for three profile face images taken from UND database. It shows the original input images, profile face edge maps approximated with lines, edge connectivity graph and ear detection results.

Figure 2.11a shows the ear detection results for Data Set 1 of IITK database which contains normal frontal ear images. To show the rotation (pose) and scale invariance of the technique, Data Set 2 of IITK database is used. Figure 2.11b gives few results from Data Set 2 where ears of different sizes and rotations are efficiently detected without any user intervention and change of parameters. The technique has also detected ears successfully for the images of Data Set 3 of IITK database (where images contain out-of-plane rotations) even for the extreme poses ($-40°$ and $+40°$). Figure 2.12a shows detection results for few images taken from IITK Data Set 3. Further, few ear localization results for extreme poses ($-40°$) where ear localization is found to be very challenging are shown in Fig. 2.12b. The technique has localized ears precisely for almost all extreme cases. It has also detected ears of all shapes (viz. round, oval, triangular, rectangular) successfully.

Table 2.2 summarizes ear detection results for IITK database. It is seen that accuracy for Data Set 1 is the highest as it contains frontal ear images. In such images, full ear structure is visible and good amount of edges are obtained which help in achieving strong connectivity among the edges representing ear. Accuracy for Data Set 2 is comparable with that of Data Set 1, in spite of images having variations in scale and rotation. This is due to the fact that the presented technique exploits the structural details of the ear which do not change with scale and rotation. Data Set 3

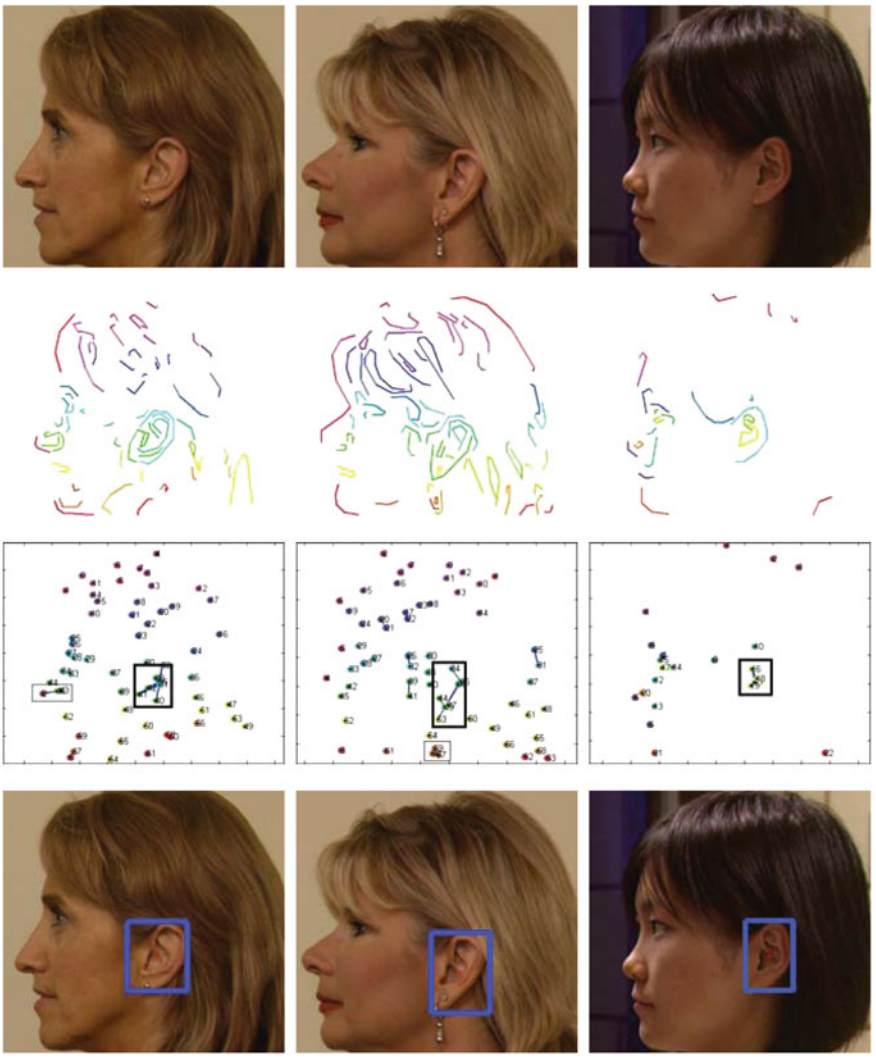

Fig. 2.10 Ear detection: (row-1) original input images, (row-2) edge maps approximated with lines (colors used to distinguish edges), (row-3) edge connectivity graphs (graph components having average vertex degree >1 enclosed in rectangles), (row-4) ear detection results

shows the least accuracy among all data sets of IITK database. This is because in the presence of out-of-plane rotation, the availability of the structural details of the ear decreases as camera moves away from the frontal position. Ear localization results for IITK database are also compared in Table 2.2 with the results reported in [33]. It is evident that the presented technique performs much better than the technique discussed in [33]. This improvement is achieved due to following reasons.

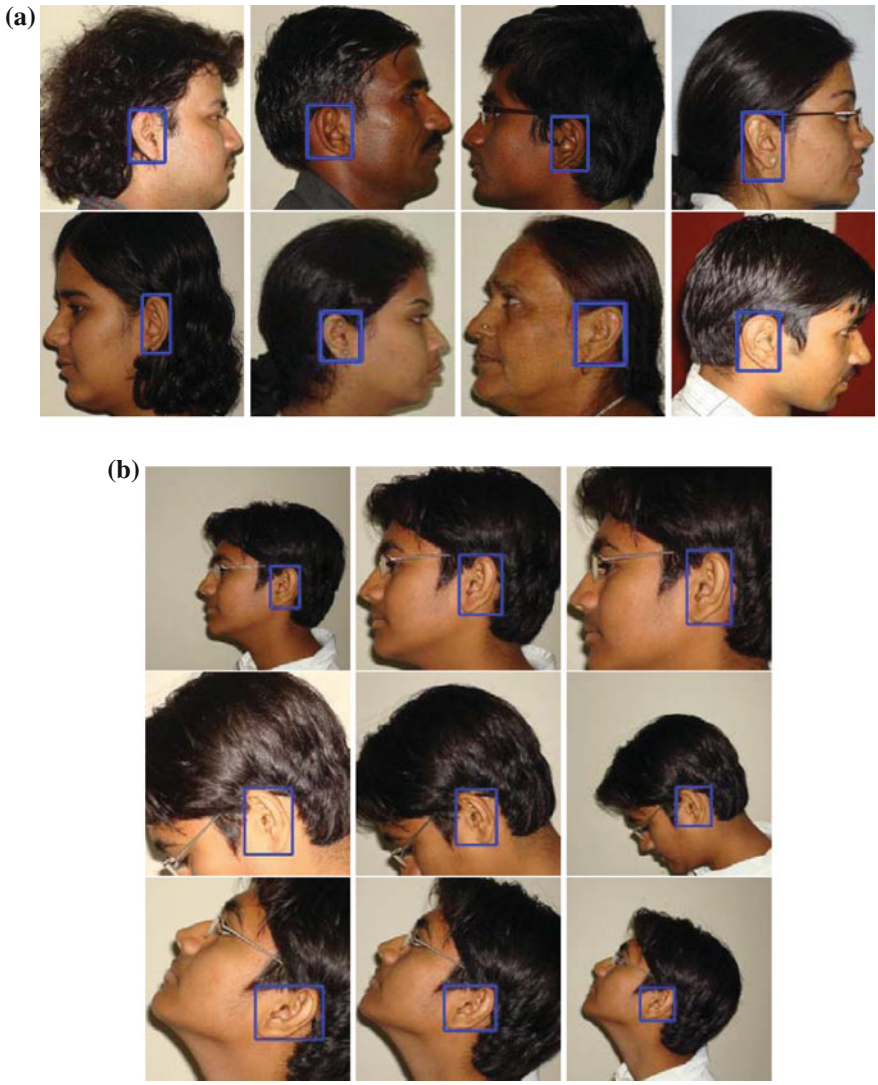

Fig. 2.11 Ear detection results for IITK database. **a** Data set 1, **b** data set 2

1. This technique breaks the derived edges of the profile face into a set of convex edges to reduce the participation of noisy edges in the cluster of true ear edges.
2. This technique has used a rotation, scale and shape invariant ear template which depicts the characteristics of ears of various scales, rotations and shapes. Identification of the true ear is performed by comparing the probable ear candidates with the ear template. Use of rotation, scale and shape invariant ear template greatly helps in localization of ears of various poses, scales and shapes.

Fig. 2.12 Ear detection results for IITK database (data set 3). **a** Detection results for two subjects, **b** detection in extreme Views

Table 2.2 Percentage accuracy for IITK database

Data set	# of test images	Ear localization accuracy (%)	
		Reported in [33]	Method discussed here
Data set 1	801	95.88	**99.25**
Data set 2	801	94.73	**98.50**
Data set 3	1070	91.11	**95.61**

3. Identification of true ear among the probable ear candidates with the help of an ear template results into much better and robust ear localization and reduces false positives. The technique in [33] performs ear localization merely based on the size of the connected components which often leads to wrong ear localization as there may exist a cluster of the largest size of non-ear edges.
4. The performance obtained in this technique is found to be robust and stable on a larger data set as compared to [33].

Ear detection results for few profile face images of University of Notre Dame (UND) database are shown in Fig. 2.13 whereas overall localization accuracies for the same database is given in Table 2.3. Ear localization accuracy for UND database

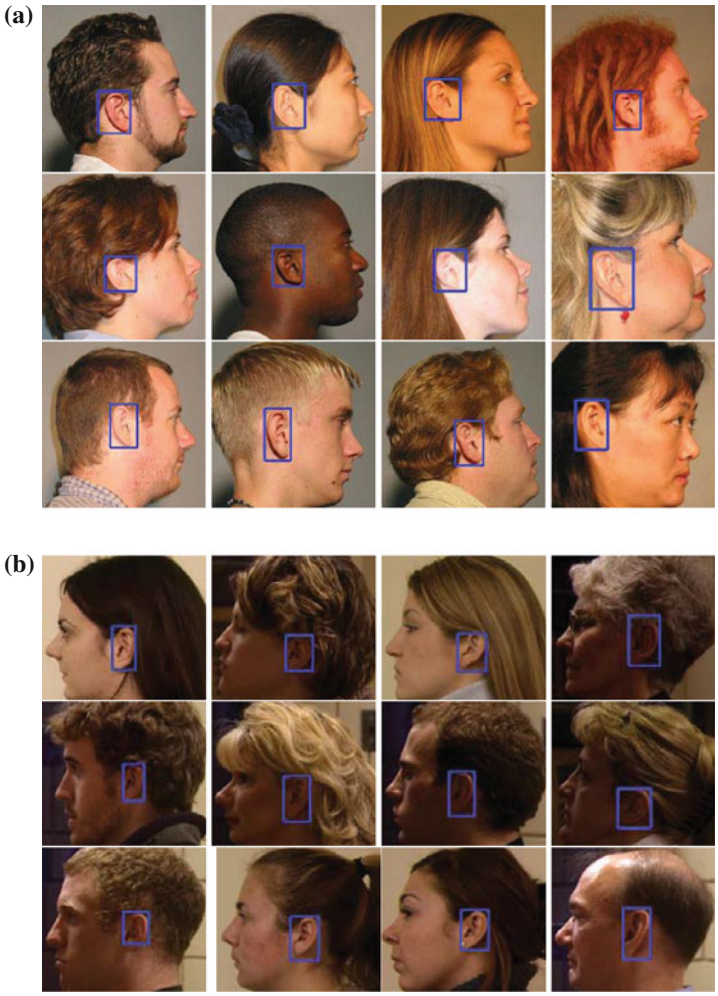

Fig. 2.13 Ear detection results for UND database. **a** UND-E data set, **b** UND-J2 data set

is found to be less as compared to IITK database due to following reason. Hair color of many subjects in UND database is similar to their skin color. Since strength of the discussed technique is derived from the successful detection of skin regions, similarity of the hair color with skin reduces the performance of skin segmentation and in turn, affects the ear localization accuracy and increases false positives.

Table 2.3 also shows comparative performance of some well known techniques on UND database. It is seen from the table that [9] produces low detection rate as compared to the technique discussed in this chapter. Moreover, it makes the assumption that the ear is the principal elliptical shape in the image which limits its use to the controlled environment and frontal ears, as the presence of background objects

Table 2.3 Percentage accuracy for UND database

Technique	Data set	# of test images	Localization accuracy (%)
[9]	Part of UND-J2	942	91
[17]	Part of UND-J2	203	100
Technique discussed here	UND-J2	1780	**96.63**
	UND-E	464	**96.34**

or posed ear may lead to false detections. The technique discussed in [17] achieves good detection rate, but the size of the test data set is very small (only 203 images). Also, if the test ear images are rotated or their appearances are changed with respect to training data, the presented technique may fail because the training images may not include such cases. Forming a database of ears with all possible rotation demands very large space and practically not feasible. Also to detect the ears of different scale, the technique should perform an exhaustive search with filters of various sizes which is computationally very expensive and makes the technique infeasible for real applications. On the other hand, the technique discussed in this chapter can inherently handle rotation (pose) and scale changes and does not incur any extra computational overhead to achieve this. Also, it is tested on a very large data set of 4916 images comprising of rotated (in-plane and out-of-plane) and scaled images which dictates the stability and robustness of the technique. A detailed comparison of [17] with the presented technique is given in Table 2.4.

Table 2.4 Comparison with the technique discussed in [17]

Parameters	Techniques	
	[17]	Presented technique
Time per detection (same configuration)	26.40 s	7.95 s
Training overhead	More. To train classifiers with 1000s of positive and negative samples	Very less. Only required to learn skin parameters and ear template using few 100 samples
Invariant to		
(i) Rotation	No	Yes
(ii) Scale	No	Yes
(iii) Occlusion	Up to some extent	No
Total test data size	Very small (307 images)	Large (4916 images)
Test data	No scaling, minor pose variation	Good amount of scaling and rotation (IITK data sets 2 and 3)

Fig. 2.14 Few failure cases from IITK and UND databases

Performance of the presented technique could not be compared with [4] because of the non-availability of the test results. Also comparisons could not be made with [19, 20] as these techniques have used XM2VTS database [10] which is not available. However, it can be noted that XM2VTS database is relatively easy to work because it contains images captured in plane background with controlled illumination and comprises of good quality images whereas UND images contain non-uniform cluttered background, poor illumination and pose variations.

The presented technique has failed to detects ears fully or partially in some cases of IITK and UND databases. Failure has occurred when ears are occluded by hair or affected by noise and poor illumination. Few examples of failure in detecting ears due to these reasons are shown in Fig. 2.14.

References

1. Burge, Mark and Burger, Wilhelm 2000. Ear biometrics in computer vision. In *Proceedings of International Conference on Pattern Recognition (ICPR'00)*, vol. 2, 822–826.
2. Hurley, David J., Mark S. Nixon, and John N. Carter. 2005. Force field feature extraction for ear biometrics. *Computer Vision and Image Understanding* 98(3): 491–512.
3. Yan, Ping, Kelvin W. Bowyer. 2005. Empirical evaluation of advanced ear biometrics. In *Proceedings of International Conference on Computer Vision and Pattern Recognition-Workshop*, vol. 3, 41–48.
4. Alvarez, L., E. Gonzalez and L. Mazorra. 2005. Fitting ear contour using an ovoid model. In *Proceedings of IEEE International Carnahan Conference on Security Technology (ICCST'05)*, 145–148.
5. Kass, M., A. Witkin, and D. Terzopoulos. 1988. Snakes: Active contour models. *International Journal of Computer Vision* 1(4): 321–331.
6. Yan, Ping, and K.W. Bowyer. 2007. Biometric recognition using 3D ear shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(8): 1297–1308.
7. Ansari, Saeeduddin and Phalguni Gupta. 2007. Localization of ear using outer helix curve of the ear. In *Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07)*, 688–692.
8. Yuan, Li, Zhi-Chun Mu. 2007. Ear detection based on skin-color and contour information. In *Proceedings of International Conference on Machine Learning and Cybernetics (ICMLC'07)*, vol. 4, 2213–2217.
9. Arbab-Zavar, Banafshe and Mark S. Nixon. 2007. On shape-mediated enrolment in ear biometrics. In *Proceedings of the 3rd International Conference on Advances in Visual Computing—Volume Part II*, 549–558.

10. Messer, K., J. Matas, J. Kittler, J. Lttin and G. Maitre. 1999. XM2VTSDB: The extended M2VTS database. In *Proceedings of 2nd International Conference on Audio and Video-based Biometric Person Authentication*, 72–77.
11. Sana, Anupam, Phalguni Gupta, Ruma Purkait. 2007. Ear biometric: A new approach. In *Proceedings of International Conference on Advances in Pattern Recognition (ICAPR'07)*, 46–50.
12. Prakash, Surya, Umarani Jayaraman and Phalguni Gupta. 2008. Ear localization from side face images using distance transform and template matching. In *Proceedings of IEEE International Workshop on Image Proceedings Theory, Tools and Applications (IPTA'08)*, 1–8.
13. Prakash, Surya, Umarani Jayaraman and Phalguni Gupta. 2009. A skin-color and template based technique for automatic ear detection. In *Proceedings of International Conference on Advances in Pattern Recognition (ICAPR'09)*, 213–216.
14. Attarchi, S., K. Faez and A. Rafiei. 2008. A new segmentation approach for ear recognition. In *Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems*, 1030–1037.
15. USTB Database, University of Science and Technology Beijing. http://www.ustb.edu.cn/resb/.
16. Carreira-Perpinan. 1995. Compression neural networks for feature extraction: Application to human recognition from ear images. Master's thesis, Faculty of Informatics, Technical University of Madrid, Spain.
17. Islam, S.M.S., M. Bennamoun and R. Davies. 2008. Fast and fully automatic ear detection using cascaded adaboost. In *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV'08)*, 1–6.
18. Prakash, Surya, Umarani Jayaraman, Phalguni Gupta. 2009. Ear localization using hierarchical clustering. In *Proceedings of SPIE International Defence Security and Sensing Conference, Biometric Technology for Human Identification VI, 730620*, vol. 7306, 730620–730620-9.
19. Cummings, A., M. Nixon and J. Carter. 2010. A novel ray analogy for enrolment of ear biometrics. In *Proceedings of International Conference on Biometrics: Theory, Applications and Systems (BTAS'10)*, 1–6.
20. Ibrahim, Mina I.S., Mark S. Nixon and Sasan Mahmoodi. 2010. Shaped wavelets for curvilinear structures for ear biometrics. In *Proceedings of 6th International Conference on Advances in Visual Computing (ISVC'10)—Part I*, 499–508.
21. Kumar, Amioy, Madasu Hanmandlu, Mohit Kuldeep and H.M. Gupta. 2011. Automatic ear detection for online biometric applications. In *Proceedings of National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG 2011*, 146–149.
22. Lankton, S., and A. Tannenbaum. 2008. Localizing region-based active contours. *IEEE Transactions on Image Processing* 17(11): 2029–2039.
23. Cai, J., and A. Goshtasby. 1999. Detecting human faces in color images. *Image and Vision Computing* 18(1): 63–75.
24. Poynton, C.A. 1996. *A Technical Introduction to Digital Video*. New York: Wiley.
25. Garcia, C., and G. Tziritas. 1999. Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia* 1(3): 264–277.
26. Bay, Herbert, Tinne Tuytelaars and Luc Van Gool. 2006. SURF: Speeded up robust features. In *Proceedings of 9th European Conference on Computer Vision (ECCV'06)*, 404–417.
27. Bay, Herbert, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110(3): 346–359.
28. Lowe, David G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.
29. Mikolajczyk, Krystian, and Cordelia Schmid. 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(10): 1615–1630.
30. Hopcroft, John, and Robert Tarjan. 1973. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM* 16(6): 372–378.
31. Bustard, J.D., and M.S. Nixon 2008. Robust 2D ear registration and recognition based on SIFT point matching. In *Proceedings of International Conference on Biometrics: Theory, Applications and Systems (BTAS'08)*, 1–6.

32. University of Notre Dame Profile Face Database, Collections E and J2. http://www.nd.edu/cvrl/CVRL/DataSets.html.
33. Prakash, Surya, Umarani Jayaraman, Phalguni Gupta. 2009. Connected component based technique for automatic ear detection. In: *Proceedings of 16th IEEE International Conference on Image Processing (ICIP'09)*, 2741–2744.

http://www.springer.com/978-981-287-374-3

Ear Biometrics in 2D and 3D
Localization and Recognition
Prakash, S.; Gupta, P.
2015, XVI, 116 p. 56 illus., 45 illus. in color., Hardcover
ISBN: 978-981-287-374-3