J. Vis. Commun. Image R. 27 (2015) 28–43

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Automatic target image detection for morphing ☆

Jaladhi P. Vyas [a,*], Manjunath V. Joshi [a], Mehul S. Raval [b]

[a] Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat 382007, India
[b] Institute of Engineering and Technology, Ahmedabad, Gujarat 380009, India

ARTICLE INFO

Article history:
Received 9 January 2014
Accepted 15 December 2014
Available online 2 January 2015

Keywords:
3D texton
K means clustering
Chi-square distance measure
Facial features extraction
Control points detection
Target image detection
Delaunay triangulation
Image morphing

ABSTRACT

In this paper, we propose a novel approach for automatic target image detection for morphing based on 3D textons and contrast. Given the source image of a human frontal face and training images with human and animal faces, our algorithm automatically finds the target image from the database of animal images. There are three major advantages of our approach: (1) it solves the problem of manual target selection as done by the researchers in morphing; (2) automatic target detection achieves smooth transition from source to destination image; and (3) control points are automatically detected for morphing because the algorithm detects the target based on the matching features. The experiments were conducted with images of six different animals viz. cheetah, lion, deer, red fox, snow leopard and rhesus monkey. For a subjective verification, a large number of human annotations are used.

1. Introduction

Morphing is a special effect that converts one image into another through a smooth transition. Morphing is used in applications such as animation, movie production, and art. The first step of morphing is to establish the correspondence between two images using pairs of feature primitives (control points) such as mesh nodes, line segments, curves or points. The warping defines the spatial relationship between all points in two images. It is found using corresponding control points in the two images. Once both images are warped into alignment, interpolation is used to generate the intermediate images [1,2]. Disadvantage with the existing morphing techniques lies in using the manual selection for target images. Usually, the selection of the target image is user defined. However, it is better to look for automated methods in which the manual interference is minimum. The goal of this paper are: (1) automatically detect the best target image based on an appropriate criteria and (2) identify common control points in both the images in order to complete the morphing process automatically. Fig. 1 shows the block diagram of the proposed approach with three steps. First step is to find target image automatically for a given source. Then, the algorithm detects the common control points and final step completes the morphing process.

Our proposed approach for target image detection is based on 3D textons and image contrast. Textons are popularly used in texture classification [3,4]. The first operational definition of 3D texton was given by Leung and Malik [5]. They defined a 2D texton as a cluster center in the filter response space. Later, to account for 3D effects, 3D textons were also proposed by them. It was generated using Leung and Malik (LM) filter bank [5], which are cluster centers of filter responses over a stack of images with representative viewpoints and lighting. Varma and Zisserman [3] achieved better classification results by using Maximum Response (MR) filter bank instead of LM filter bank. They used neighborhood of image patch, instead of filter responses to improve the accuracy of classification [4]. Our proposed approach works as follows: in the first step, textons are generated by sorting the neighborhood pixels in an image patch at every location. Contrast defines the strength of texture pattern so, in second step, local contrast generated from neighborhood of a given image is added as an illumination dependent information. Finally, the joint histogram of textons which are generated using the training set and local contrast of a given image is used as an appropriate image model for target detection. Each class has many images captured under varying conditions with certain number of textons. The model is partially rotation invariant and our training set consist of images with different size to incorporate the effect of scale change.

☆ This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author.
E-mail addresses: vyasjaladhi@gmail.com (J.P. Vyas), manjunath.joshi@gmail.com (M.V. Joshi), mehul.raval@gmail.com (M.S. Raval).

http://dx.doi.org/10.1016/j.jvcir.2014.12.005
1047-3203/

Next step is to automatically detect the control points in the region of eyes, lips and nose for the source image. Once they are detected from the source image, the same control points are referred to in the target image, to complete morphing automatically using the approach proposed in [2]. Note that, the control points vary due to change in expression of a person or due to change in a mouth length of different persons. The proposed approach is compared with Java Content Based Image Retrieval (JCBIR) [6] image search application licensed by MIT [7] and with other feature based approaches [3,4,8,9] for image retrieval. For subjective verification of ground truth target image, Amazon Mechanical Turk [10] is used to incorporate large number of human annotations. Motivation behind the work and use of natural images for morphing is due to the following reasons. In recent animation movies, cartoon face looks more natural as human facial features are added to it. Cartoon images have piecewise linear intensities but this is not the case with natural images. Moreover, colors of the cartoon faces may not remain similar within same class of images as it depends on the cartoonist imagination. However, with the introduction of the human like features in cartoons such ambiguities are diminishing.

The flow of the paper is as follows: Section 2 reviews the related work. The proposed approach for target detection based on histogram model is discussed in Section 3. The effects of contrast, rotation and scale on the proposed model are discussed in Section 4. Section 5 describes the method of automatic region extraction and control points detection for morphing. Results of our proposed approach and conclusions are discussed in Sections 6 and 7, respectively.

2. Related work

In the prior work on morphing, meshes have been used to define the position correspondences in an image pair where, mesh coordinates are interpolated by bi-cubic splines to form a warp sequence [11]. Field morphing is used to establish line correspondence, with which warp is determined according to the distance of points from the lines [12]. A novel approach for image morphing to handle the scattered feature is discussed in [13]. Authors in [14] have discussed the idea of morphing between multiple images called as polymorphing. A technique for morphing using two homomorphic triangular meshes is discussed in [15]. Warping functions that are continuous and one-to-one which preclude folding of warped images have been discussed in [16]. The plenoptic editing method for 3D morphing of objects represented by images is discussed in [17]. Authors in [18] have suggested light field morphing using 2D features. A novel approach of expression modeling and morphing based on a geometry-based paradigm is discussed in [19]. A new technique for modeling textured 3D face is briefly discussed in [20]. An approach to recover the face position and the facial expression automatically from the video sequence using 3D morphing is discussed in [21]. An optimal morphed field is generated by changing the brightness and geometry in [22]. The method for face structure extraction and recognition using 3D morphing is discussed briefly in [23] and based on it the new approach for face recognition has been proposed. Unlike other approaches as discussed above, the proposed work on morphing chooses the best matching target image automatically.

Control points detection is an important pre-processing step in morphing. Many techniques have been suggested to detect the control points automatically using both source and target images. A new approach for feature specification using snakes and warp generation is discussed in [24]. Authors in [25] have suggested a new approach to detect the features using active shape model and then perform morphing automatically. Our method to detect the control points is different from these earlier approaches. In that, control points are automatically detected for human as well as animal face images. Control points detection method is divided in two parts: (1) detect two control points of eyes in both source and the target image and (2) other control points are detected using geometric distance relationship known for human and animal face. In this paper we focus on the problem of simultaneous target detection and morphing, and to the best of our knowledge, this problem has not been attempted by the researchers so far.

3. Proposed approach

Automatic target image detection is the first step of the proposed approach. It consist of three stages with training set used in stage 1 and the test images in stage 2 and 3 respectively. The details are discussed below.

Stage 1. Generating the texton dictionary: The texton dictionary is generated from the training set which includes color face images of human and six different categories of animals viz. cheetah, lion, deer, red fox, snow leopard and rhesus monkey. Here, we consider each of these seven (human and six animals) different sets as a class with same number of training images. Each image in the training set is converted to a gray scale. Then, at each location of an image, eight neighbors of 3×3 patch are selected to form a vector of length 8×1. These vectors are then sorted either in ascending or descending order. With $N \times N$ pixels in each image, we get N^2 vectors each of size 8×1. The process of collecting eight neighborhood pixels for every 3×3 patch leads to eight output images. Steps to generate sorted vector from 3×3 patch are shown in Fig. 2. The input image and eight output images generated by sorting eight neighbors is shown in Fig. 3. Let T_{img} indicates the number of training images in a class. Then eight output images are stacked in such a way that 8×1 vector at every pixel location becomes a new vector of size $8T_{img} \times 1$. This process is shown in Fig. 4. Suppose the number of training images in a class i.e. $T_{img} = 7$. Then at each pixel position, we have a vector of size $8T_{img} \times 1$ i.e. 56×1 (as $T_{img} = 7$). Considering $N \times N$ pixels for each image, the total number of vectors remains N^2 each of size $8T_{img} \times 1$ i.e. 56×1. These vectors are clustered using *K-means* algorithm. Resulting cluster centers are known as 3D textons. The same procedure is repeated on images of every class and each class of images leads to K number of textons. Textons from all classes are combined to form a dictionary. Here, dictionary is a matrix whose columns are textons from all the classes. If N_{class} represents the number of classes then, there are $N_{class}K$ number of textons

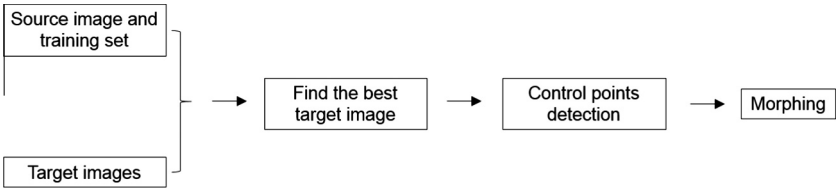

Fig. 1. Proposed approach.

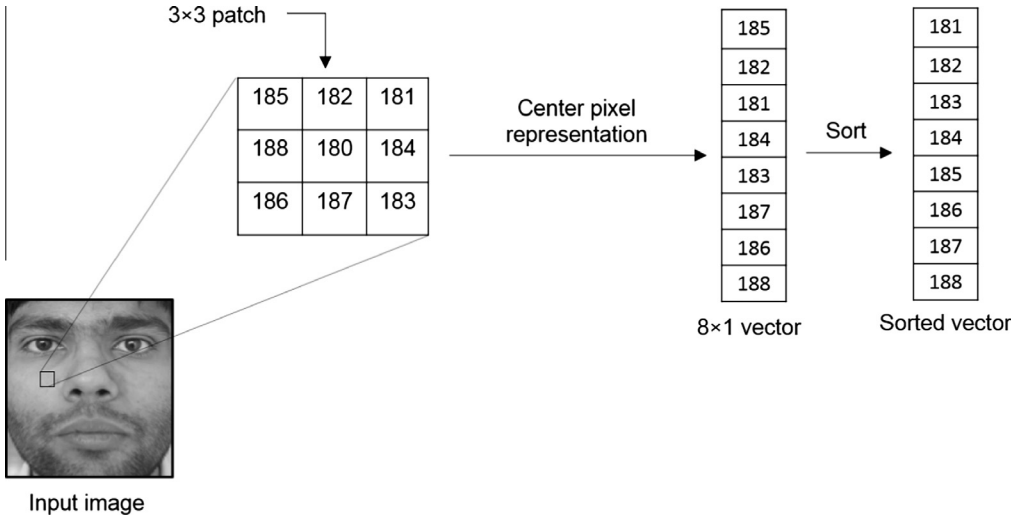

Fig. 2. The process to generate 8×1 vector from 3×3 patch.

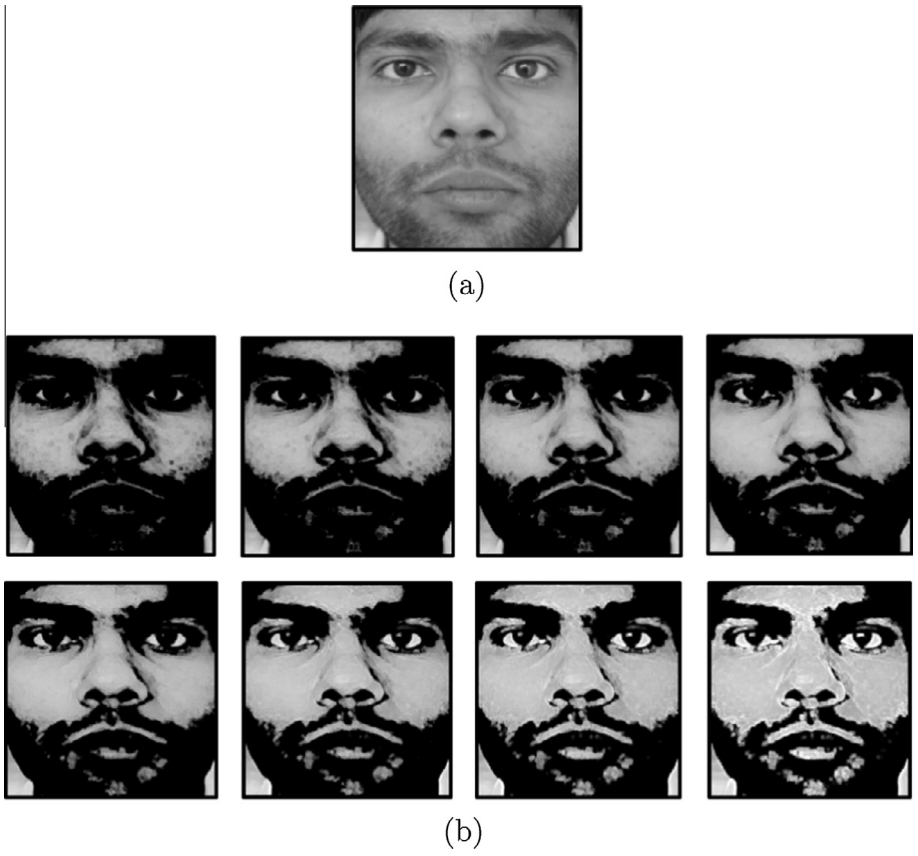

Fig. 3. (a) Input image and (b) eight output images generated by sorting eight neighbors of every 3×3 patch.

each of size $8T_{img} \times 1$. Therefore, the dictionary dimension is $8T_{img} \times N_{class}K$. For example, if we choose the value of $K = 5$ in K-means algorithm i.e. total number of textons per class is 5 and if $N_{class} = 7$ then, total number of textons in the dictionary are $7 \times 5 = 35$. If the size of each texton is 56×1 then, the dictionary dimension is 56×35. Fig. 4 shows the necessary steps of generating K textons for one class.

Stage 2. Histogram based model generation: The histogram model generation involves local contrast and textons. The model is generated for a given image using textons with the help of training set and local contrast of that image. The process of histogram based model creation starts by computing the local contrast from an image. To generate a local contrast for each 3×3 patch in a given image, the first step is to find two mean gray level values in a patch. One of the mean value is from pixels with gray level values above the center pixel and the other is from pixels with gray level values below the center pixel. Let us call them as A_a and A_b respectively. Then $A_a - A_b$ gives the contrast at the center pixel

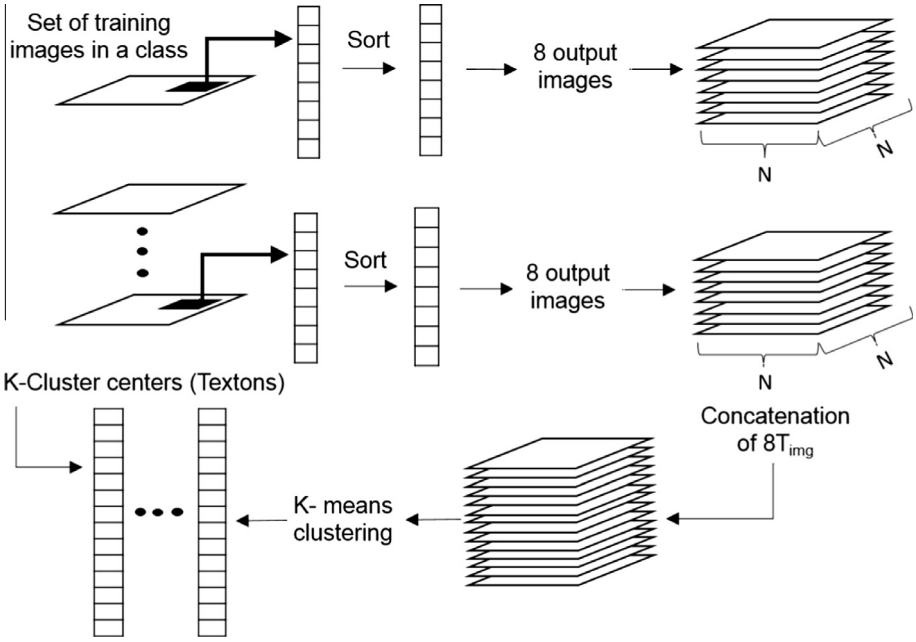

Fig. 4. Steps to generate the K textons from one class.

location. The contrast image is generated by repeating the above steps for all pixel locations. We further divide the values of local contrast into fix number of bins. An example to calculate the contrast is shown below for a 3×3 image patch [26]. Note that, texton is a vector quantity generated from the training set while local contrast is a scalar generated from each 3×3 patch of a given image.

5	7	11
20	10	20
15	2	7

A 3×3 patch with center pixel value = 10

Contrast $C = \frac{20+20+15+11}{4} - \frac{5+7+2+7}{4} = 16.5 - 5.25 = 11.25$

The second feature used to generate the histogram model is texton. Here, the given image is first converted to a set of 8×1 vectors. If $N \times N$ is the size of an image then, the total number of vectors are N^2. Let N_{tex} represents the total number of textons in the dictionary and N_{bin} represents total number of bins for local contrast then, the joint histogram model has $N_{tex} \times N_{bin}$ number of bins. Now for a given image the joint histogram model is generated using vector–scaler combination of textons and local contrast as follows: From the local contrast image, at each pixel location we find its corresponding bin value from N_{bin} bins. Then, for each of the 8×1 vectors, Euclidean distance with all textons (N_{tex}) are computed and the texton at the smallest Euclidean distance is chosen. Using the above steps, corresponding bin count of total $N_{tex} \times N_{bin}$ bins of the joint histogram is increased. The joint histogram model is generated by repeating the above steps for all pixels in a given image. Note that, the size of image vector is 8×1 and the texton size is $8T_{img} \times 1$, therefore resizing of textons in the dictionary is needed for comparing the Euclidean distance. This is done by representing each texton in the dictionary by a set of 8×1 vectors as shown in Fig. 5. As explained in the first stage, if total number of training images in a class (T_{img}) are 7 then, the size of each texton becomes 56×1. Therefore, each texton can now be represented by seven vectors of size 8×1 vectors. Note that, the same bin is considered for all the seven vectors while comparing the image vector and texton. Fig. 6 shows an input image and its corresponding joint histogram model, where $N_{tex} = 35$ and $N_{bin} = 13$.

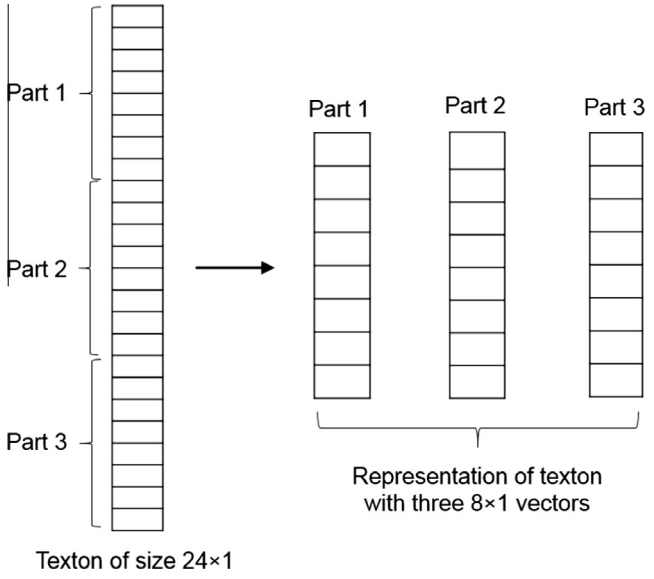

Fig. 5. Representing textons to make the size of patch vector 8×1 and texton same.

Stage 3. Target detection: The target image database has animal face images of six different classes (cheetah, lion, deer, red fox, snow leopard and rhesus monkey). Here, one may argue to use parametric test for comparison but it cannot be used due to unknown distribution of the samples. Therefore, once histogram models are available for a given source and target images, non parametric chi-square (χ^2) significance test is used as a measure of similarity between two histograms (h_1 and h_2). Equation of chi-square measure for two histograms is as below.

$$\chi^2(h_1,h_2) = \frac{1}{2}\sum_{n_1=1}^{N_{tex}}\sum_{n_2=1}^{N_{bin}} \frac{(h_1(n_1,n_2) - h_2(n_1,n_2))^2}{h_1(n_1,n_2) + h_2(n_1,n_2)}. \quad (1)$$

One can also use Kullback–Leibler (KL divergence) as a distance measure. Authors in [27] compared texture classification results using KL divergence and χ^2 statistics as distance measures and dis-

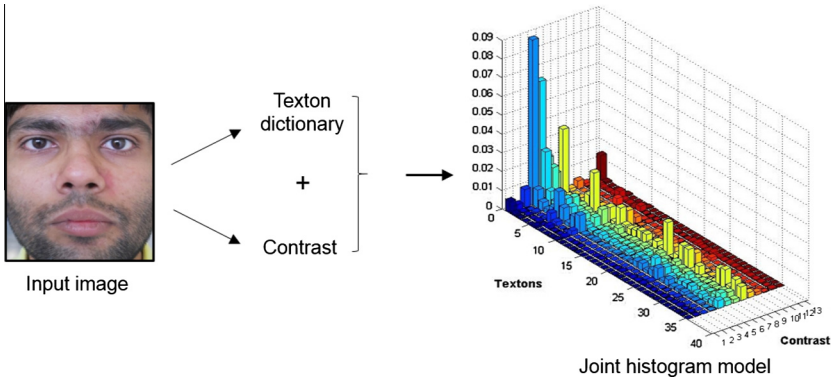

Fig. 6. Joint histogram model of an image.

cussed their theoretical pros and cons associated with these two approaches as mentioned in [27]. In the proposed approach we use chi-square distance as a similarity measure.

3.1. Effect of contrast, scale and rotation on the histogram model

In this section, we address the effects of local contrast variations, change in a size and rotation of the given image on the histogram model.

Local contrast: Contrast is the property of an image, and also an essential cue for human visual system. The local contrast depends on gray scale values of an image patch and ranges in between $[-255,255]$. However, its distribution is not uniform in many images as large number of values lie close to zero. Therefore, fixed length binning of the joint histogram will not resist contrast variations. This motivates us to use variable length binning which spreads the joint histogram. The variable length binning is carried out as follows: The interval closer to zero is assigned large number of bins, and the remaining portion is assigned smaller number of bins with total number of bins remaining constant i.e. N_{bin}. It is interesting to note that, this type of non uniform quantization is used in pulse code modulation (PCM) of speech samples. Fig. 7 shows the histograms of local contrast using fixed and variable length binning for two different class of images. One can note, the variations in the histogram is less for fixed length binning as compared to variable length binning. This indicates that variable length binning is useful for separating interclass images.

Rotation invariance: There are several ways to achieve rotation invariance as discussed in [4,28,29]. We have incorporated the effect of image rotation in the histogram model by using two rotational invariant features: textons and local contrast. As discussed in stage 1 of the proposed approach, textons are generated by extracting every 3×3 patch from all images in the training set and by converting them to sorted vectors of length 8×1. From this it is clear that, the sorted vectors generated from 3×3 patch remain unaltered in spite of image rotation. Therefore, the texton generated from these sorted vectors is invariant of image rotation. The second feature, i.e. local contrast generated by subtracting two mean gray level values which are above and below the center pixel value. Hence this is also rotation invariant as mean gray values remain same in spite of rotation. For the original and rotated image, we illustrate these facts through an example. Two 3×3 patches with center pixel value of 10 is shown. Here, the local contrast $C = \frac{20+20+15+11}{4} - \frac{5+7+2+7}{4} = 16.5 - 5.25 = 11.25$ is same for rotated and un-rotated 3×3 patch.

5	7	11
20	10	20
15	2	7

$\xrightarrow{90^o rotation}$

15	20	5
2	10	7
7	20	11

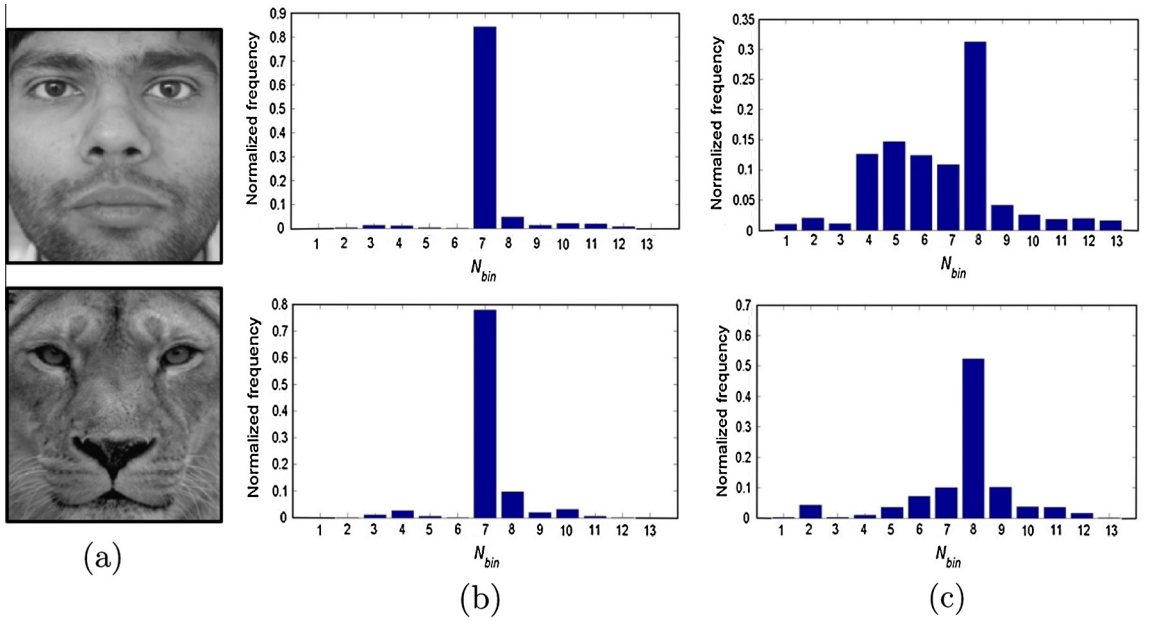

Fig. 7. (a) Input gray scale images, (b) histograms of local contrast with fixed length binning and (c) histogram of local contrast using variable length binning.

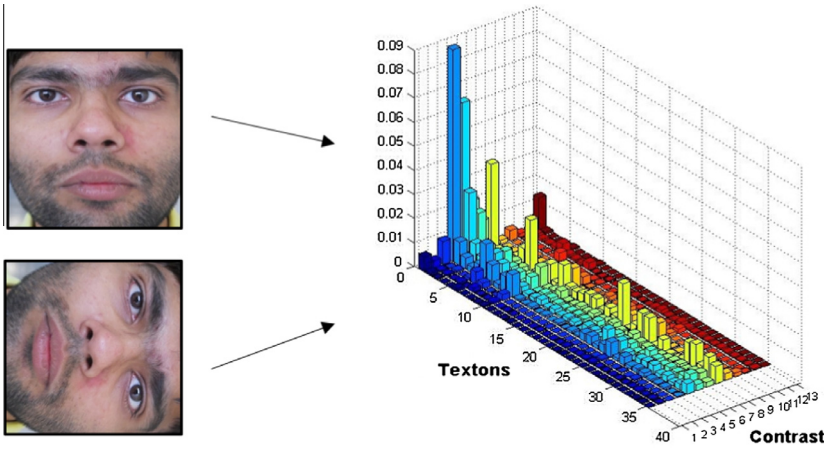

Fig. 8. Rotation invariant image model.

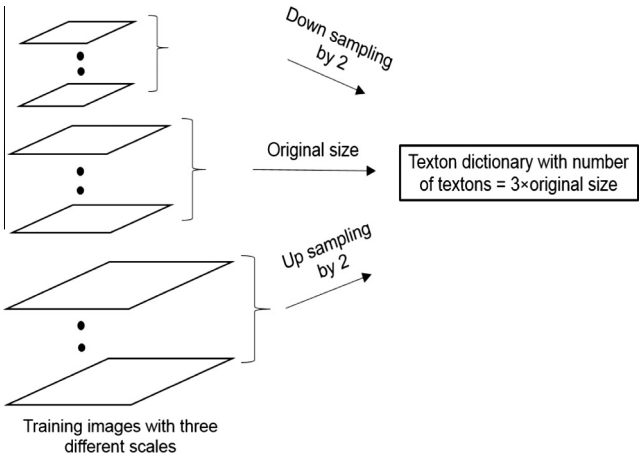

Fig. 9. Images with three different scales in the training set to achieve partial scale invariance.

Consequently, the joint histogram model of textons and local contrast is invariant to image rotation. However, one must note that, certain angles of rotation yield an interpolated value since the rotated pixel location may fall outside the integer image grid. Hence rotated and unrotated patch values differ for such cases. Therefore, the histogram model generated using the above steps is partially rotation invariant due to interpolation effects. For a given image its rotation invariant image model is shown in Fig. 8. It indicates that, same model is obtained when the image is rotated by angle of 90°. Interpolation is not required for such cases as the rotation leads to an integer pixel location.

Effect of scale change: To incorporate the effect of scale change, we extended the training set by including images with three different scales. These three scales are: an original scale, down sampled by a factor of two, and up sampled by a factor of two, respectively. Textons are generated separately using these three scales of the training images. For each texton generated from the original scale, a nearest texton from other two different scales are selected and a set of 3 textons is formed i.e. textons from different scales are used to form groups. Hence, the number of sets now represents the total number of textons. Now, instead of comparing each 8×1 vector with its nearest texton, we use its nearest set. In the texton generation step, if $K = 5$ then, the same value of K is used to generate textons for other two scales in a class. Effectively, the size of texton dictionary is increased by a factor of 3. This is shown in Fig. 9. This way, we incorporate the effect of scale change. One may note that, when the image is scaled up or down, a small amount of contrast change occurs due to the use of interpolation. Therefore, the proposed model is not completely scale invariant. Fig. 10 shows histograms of the local contrast for 200×200 and 100×100 size images using variable length binning. From the difference in the distributions of the two histograms, it is evident that scale change affects the histogram albeit partially.

4. Automatic region extraction and control points detection

After detecting the target image, the next step is to automatically detect the regions of interest (ROI) and obtain the control

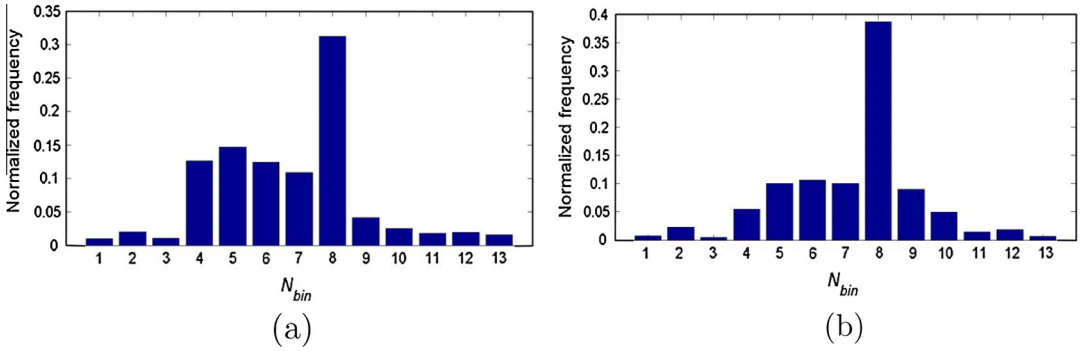

Fig. 10. Distribution of local contrast using variable length binning for (a) 200×200 and (b) 100×100 size images, respectively.

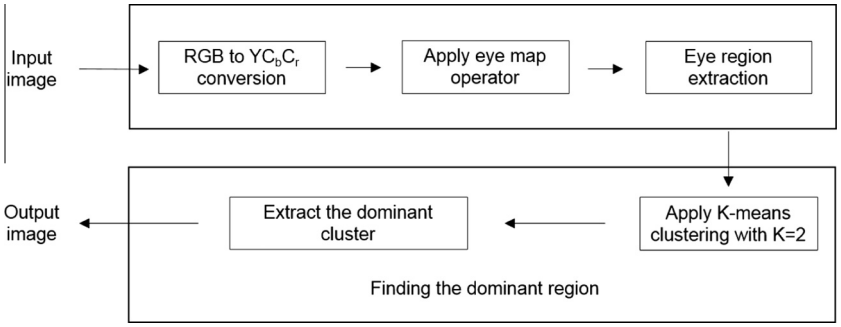

Fig. 11. Block diagram of eye region extraction.

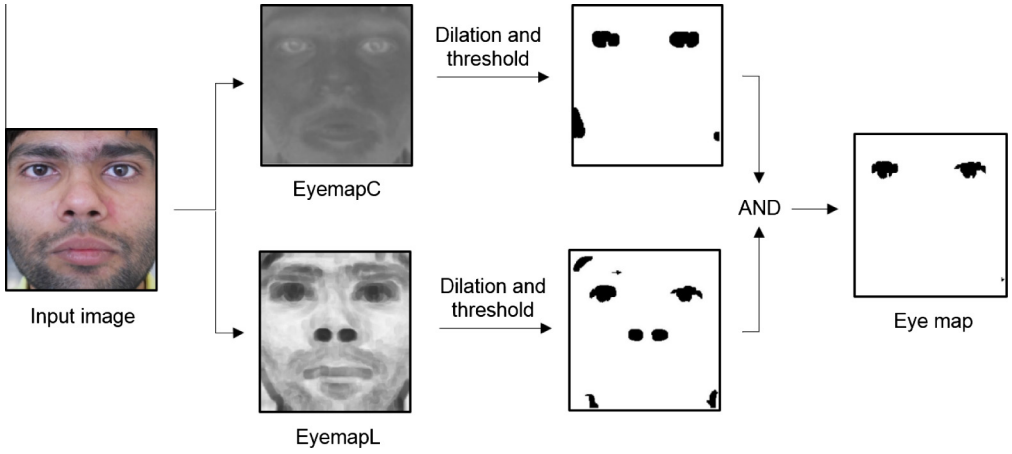

Fig. 12. Necessary steps to generate the eye map.

Fig. 13. Output of eye region extraction for a human face image.

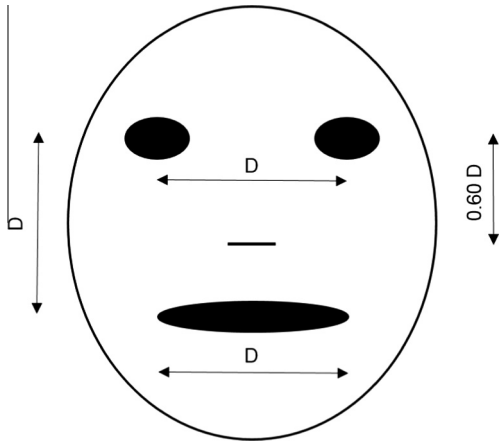

Fig. 14. Geometric human face model [36].

points. Many techniques have been suggested to detect the control points [30–34]. However, these approaches are template based which requires a priori assumptions about orientation of the face. Authors in [35] locate the human face from color images by detecting eye and mouth features automatically.

In the proposed approach, first we detect the eyes in both source and the detected target animal face using eye map operator [35] and *K means clustering*. Once the two eyes are located, more control points are found using geometric distance relationship

Table 1
Geometric distance relationship between control points for human and animal face images with respect to the distance D between two eye control points.

Distances	Human face	Animal faces of database
Eye–nose tip	$0.60 \times D$	$0.65 \times D$
Eye–lips	D	$1.30 \times D$
End points of lips	D	D

between various regions in human and animal face. One may find spatial similarity in the eyes of humans and animals while other features such as nose and lips have different shapes with multiple colors and textures. Therefore, eyes are used as a common feature for automatic detection of the other control points in both source and target image. Control point localization is aided by the invariant signatures of face i.e. natural parallelism between two imaginary line segments; first line passing through two center points of the eyes and second line between two end points of the lips.

Here, the eyes are detected using eye map operator [35] and *K means clustering*. As a first step in ROI detection, we convert input *RGB* image into YC_bC_r space. Then two eye maps are built, one for *luminance* and other for *chrominance* component, which are then combined to form a single eye map. The eye map from *chrominance* component is given by

$$EyemapC = \frac{1}{3}\left\{\left(C_b^2\right) + \left(\overline{C_r}\right)^2 + \frac{C_b}{C_r}\right\}. \quad (2)$$

Here $C_b^2, \left(\overline{C_r}\right)^2, \frac{C_b}{C_r}$ are normalized to the range 0–1 and $\overline{C_r}$ is negative of C_r i.e. $(1 - C_r)$ and the value of $\frac{C_b}{C_r}$ is considered as zero whenever C_r becomes zero. Gray-scale *dilation* ($\oplus$) and *erosion* ($\ominus$) with

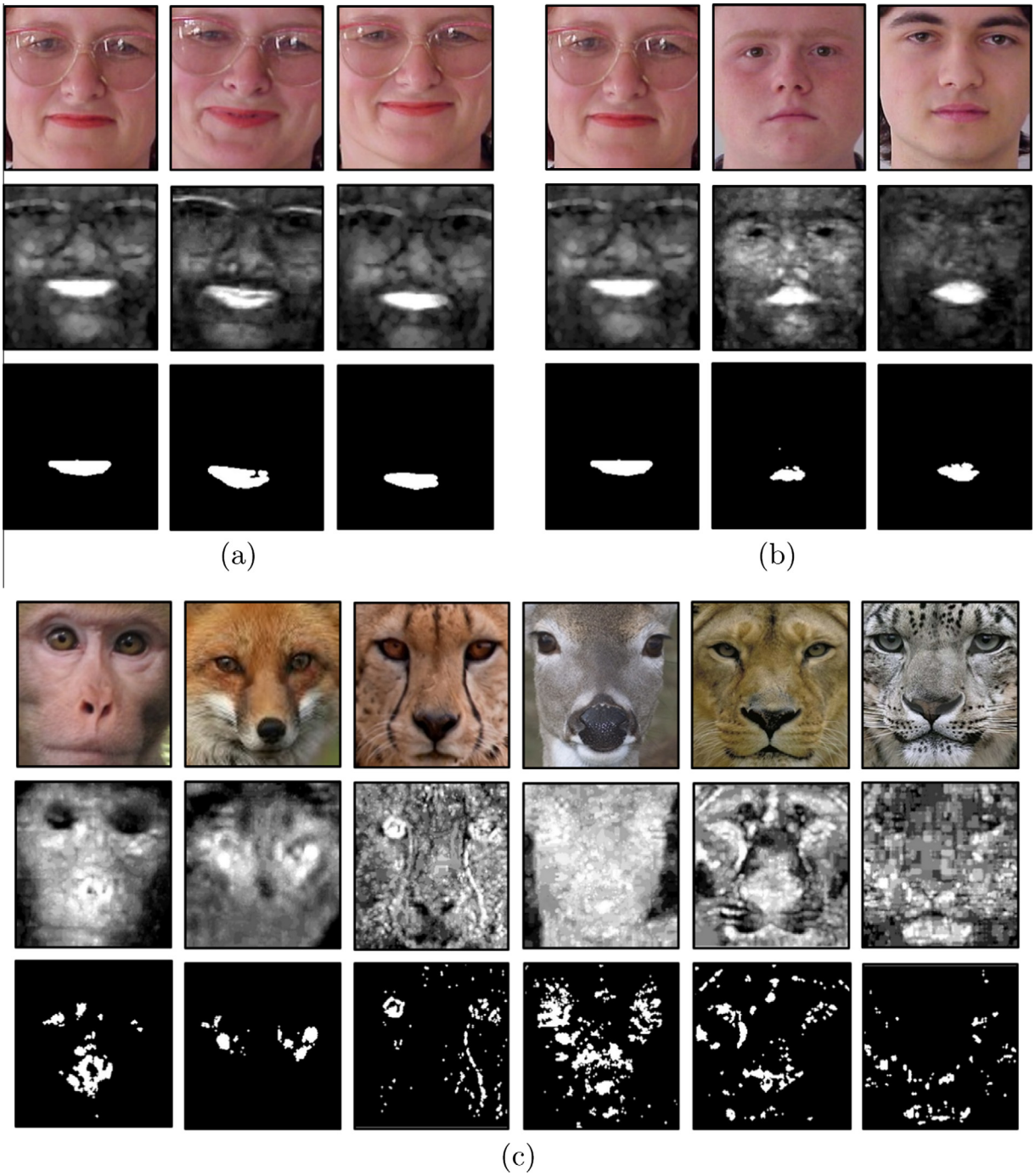

Fig. 15. Output of mouth region extraction using mouth map operator for (a) different facial expressions of a single person, (b) change in mouth length of different persons and (c) different animal face images.

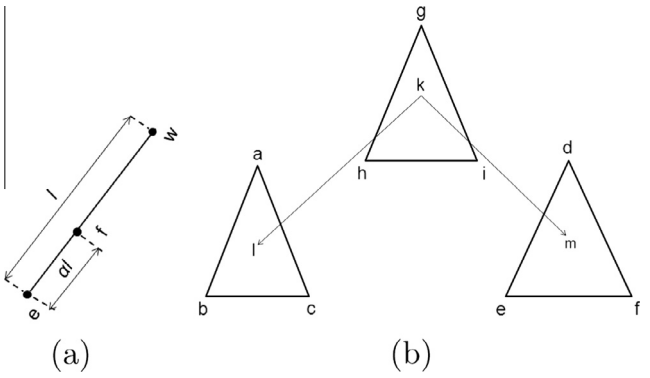

Fig. 16. (a) Process to find the position of a new feature point and (b) triangle *abc* and triangle *def* are the corresponding triangles of a source and target image respectively. Triangle *ghi* is generated by finding new positions of the feature points. *k* is one of the internal points of a new triangle *ghi*, where *i* and *m* are its corresponding points of triangles *abc* and *def* respectively.

hemispheric structuring element is used to construct the eye map for the *luminance* component as

$$EyemapL = \frac{Y(x,y) \oplus g(x,y)}{Y(x,y) \ominus g(x,y) + 1}, \quad (3)$$

where $Y(x,y)$ is the luminance component of the face region and $g(x,y)$ is a structuring element. Morphological dilation and thresholding operations are applied on the images generated using *EyemapC* and *EyemapL* maps. These images are combined by *AND* operation to generate the final eye map.

While detecting the eye regions, other noisy regions with different intensity values are also extracted. Therefore, the output image is cleansed by using *K means clustering* with $K = 2$. Fig. 11 shows the block diagram of eye region extraction. The process to generate the eye map is shown in Fig. 12. Fig. 13 shows output eye region from human face before and after noise removal.

Once eyes are located, the next step is to detect additional control points for morphing. The control points of two eyes are the two dominant connected components as shown in Fig. 13. Then, additional five control points of human face are detected using geometric model [36] and known distance relationship as given in Fig. 14 and Table 1, respectively. Here, the distance relationship of facial features for animal faces is determined experimentally. The five control points are: nose tip, two end points of the lips, center of two detected eyes and center of two end points of the lips. The parallelism between two line segments viz. the line segment joining two eye control points and other one is connecting two end points of the lips is used as the invariant signature. This signature holds true for both human and animal face. These features are not affected by change in the pose and rotation of the face image. Thus, in total seven control points are detected automatically. Note that, the control points are not invariant with the change in facial expressions of single person and change in mouth length for different persons. This is because the control points include the mouth regions of different species (human and animal faces) and they correspond to different spatial locations in human and animal face which makes it difficult to obtain invariance to expression change and mouth length variations. To extract the mouth control points one can use the mouth map operator as discussed in [35]. Fig. 15(a) and (b) shows the results of a mouth region extraction with different expressions of a single person and change in mouth length of different persons, respectively. However, the same method is not useful to detect the mouth region of animal faces which is clearly reflected in Fig. 15(c). Therefore, instead of mouth map operator we use geometric distance relationship to detect the other control points. The advantage of our approach lies in detecting the common control points in the source and the target image even though they belong to different species i.e. person and animal.

5. Morphing

After detecting target image and control points for both source and target images, the final step is morphing. As the focus of this paper is to find the target image for morphing, it is done using the mesh based approach as discussed in [2]. It includes two steps: (1) warping and (2) cross-dissolve [1]. Based on the locations of the detected control points, the source and target images are divided

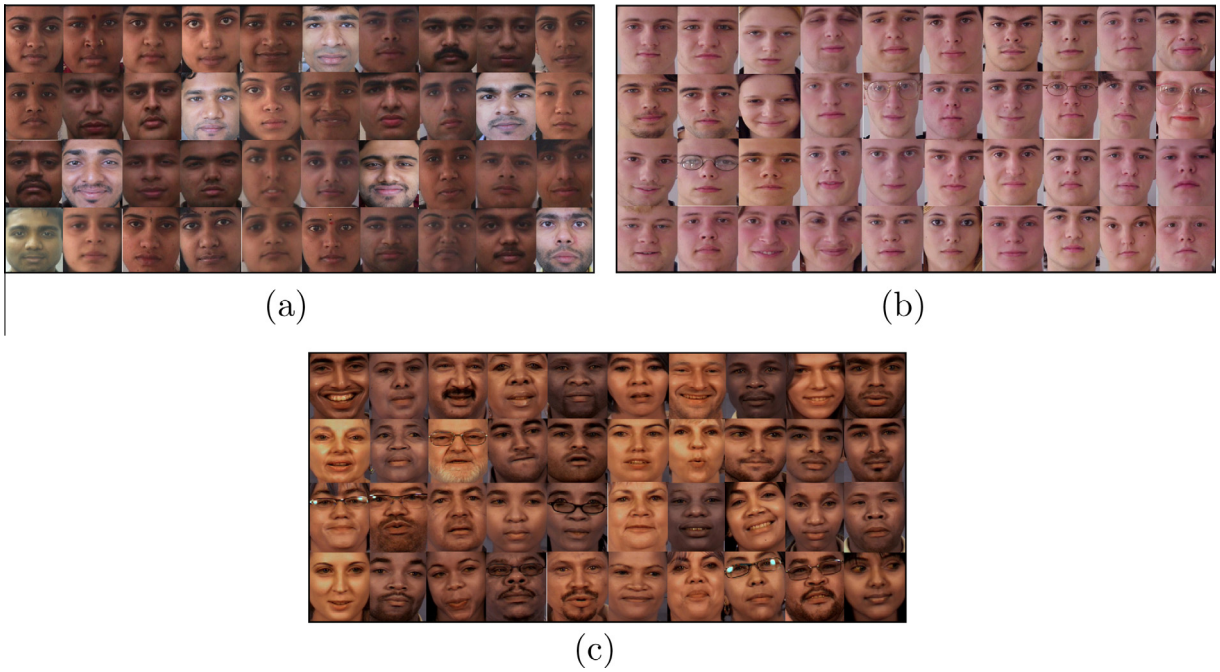

Fig. 17. Sample images from (a) Indian face database [39], (b) CVL face database [41,42] and (c) Muct face database [43].

Fig. 18. (a) Training set with seven different classes and (b) target animal face database.

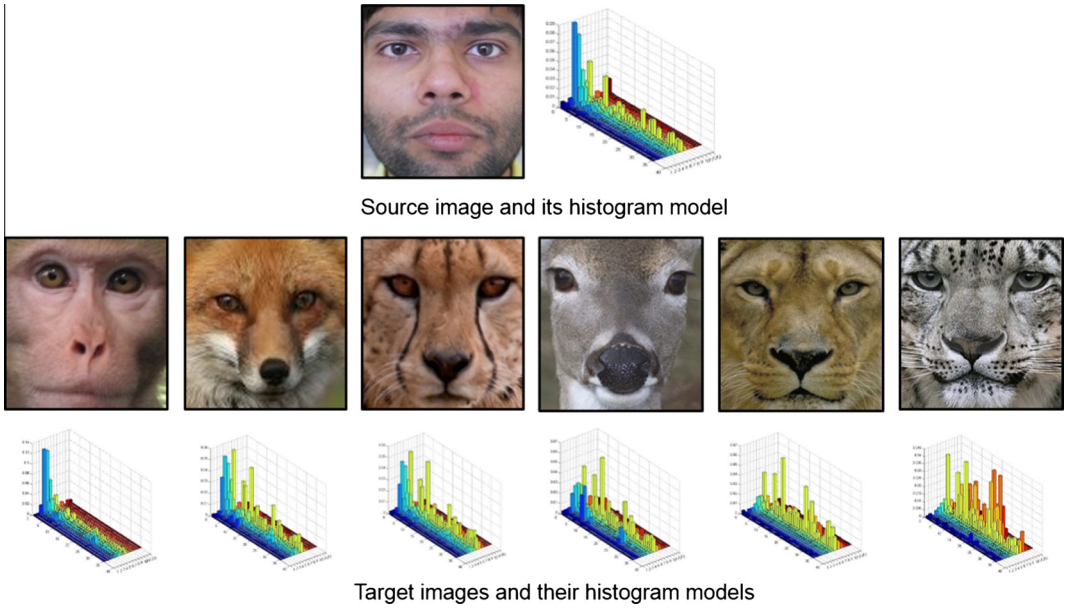

Fig. 19. Source and six different target images with their histogram models.

Table 2
Chi-square distance between source image and six different target images for three different scenarios.

Approach	Chi-square distance of six target images					
Original size (200 × 200)	0.1698	0.3060	0.3518	0.3127	0.4199	0.4468
Rotation 45°	0.1720	0.3085	0.3550	0.3175	0.4230	0.4502
Down-sampled by factor 2	0.1604	0.2975	0.3445	0.3052	0.4109	0.4392

into a fixed set of triangles using delaunay triangulation [37]. Let the weighting factor for a source image is α and for a target image is $(1-\alpha)$ respectively, and the range of α is $(0,1)$. Let e is the feature point of a source image and its corresponding feature point in the target image is w. Then the position of a new feature point f can be found out using linear interpolation as shown in Fig. 16(a). This process is repeated for all feature points in both source and target images. After completing the above steps, a new blank image with positional information about new feature points is generated. Then Delaunay triangulation is applied at the

Fig. 20. Examples of false matching using JCBIR approach.

Fig. 21. Target image retrieved using JCBIR and the proposed approach.

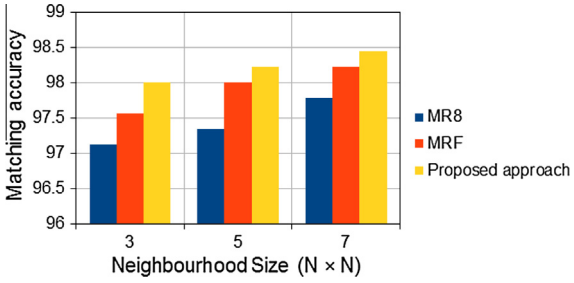

Fig. 22. Comparison of matching accuracy of three different methods with different size of neighborhood.

Table 3
Comparison of matching accuracy between JCBIR and the proposed approach.

Database	Proposed approach		JCBIR	
	$K = 5$ (%)	$K = 8$ (%)	$K = 5$ (%)	$K = 8$ (%)
Indian	98.41	95.23	96.82	93.65
CVL	98.24	96.49	97.36	93.85
Muct	97.46	94.92	95.28	93.47

Table 4
Comparison of matching accuracy for different feature based approaches.

Database	Methods				
	LTP [8] (%)	LTrP_2nd_order [9] (%)	MR8 [3] (%)	MRF [4] (%)	Proposed approach (%)
Indian	92.06	93.65	95.23	96.82	98.41
CVL	94.73	96.49	95.61	97.36	98.24
Muct	94.20	95.65	96.30	97.10	97.46

Table 5
Comparison of matching accuracy for three different scenarios.

Rotation angle (°)	MRF [4] (%)	Proposed approach (%)
0	97.57	98.01
90	96.25	98.01
45	96.02	97.79

position of new feature points. For every triangle in the new image, the value of each pixel is found from its corresponding source and target image triangles using following method: Let *abc* and *ghi* are the triangles of source and new image respectively. Suppose k is the pixel position in the triangle *ghi*. Then its corresponding pixel in triangle *abc* is found using the following equations:

$$k = \lambda_1 g + \lambda_2 h + \lambda_3 i \text{ where, } \lambda_1 + \lambda_2 + \lambda_3 = 1, \quad \lambda_i \geqslant 0 \text{ then,}$$

$$i = \lambda_1 a + \lambda_2 b + \lambda_3 c \text{ and,}$$

$$m = \lambda_1 d + \lambda_2 e + \lambda_3 f.$$

Here, all alphabets represent x and y positions of the feature points. The same process is repeated to find the corresponding pixels from the target image. Finally, in a new blank image, for every location, we have corresponding pixels from a source and target image respectively. Then a sequence of images using the cross dissolving technique is generated with the help of the following equation [38]:

$$k(x,y) = \alpha i(x,y) + (1-\alpha) m(x,y), \quad 0 \leqslant \alpha \leqslant 1. \tag{4}$$

6. Experimental results

6.1. Datasets

In order to conduct the experiments a set of frontal face images [39–43] are collected for seven different classes with varying illumination conditions and small variation in pose. Selected class of

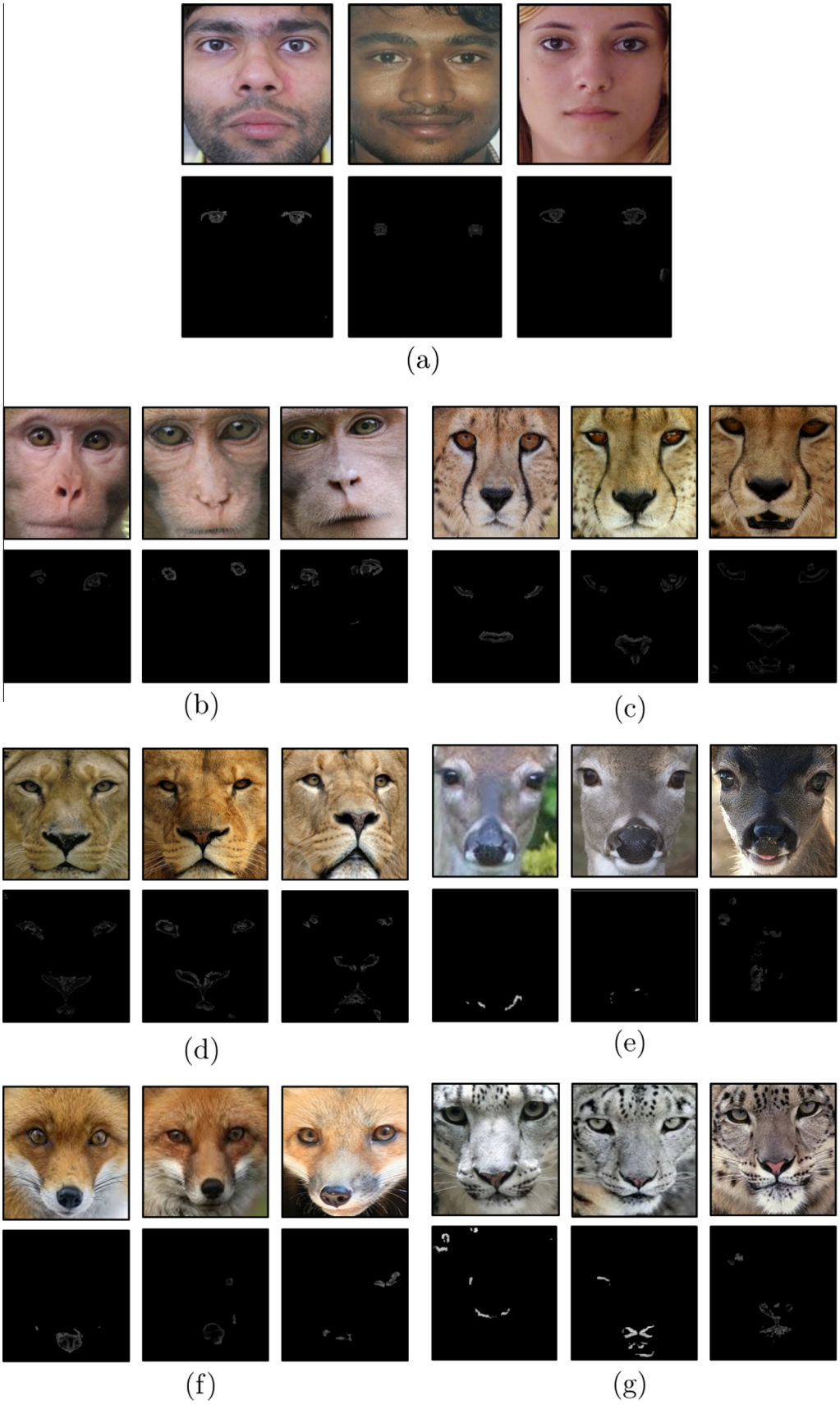

Fig. 23. Eye region extraction for different class of images. (a) Human, (b) rhesus monkey, (c) cheetah, (d) lion, (e) deer, (f) red fox and (g) snow leopard.

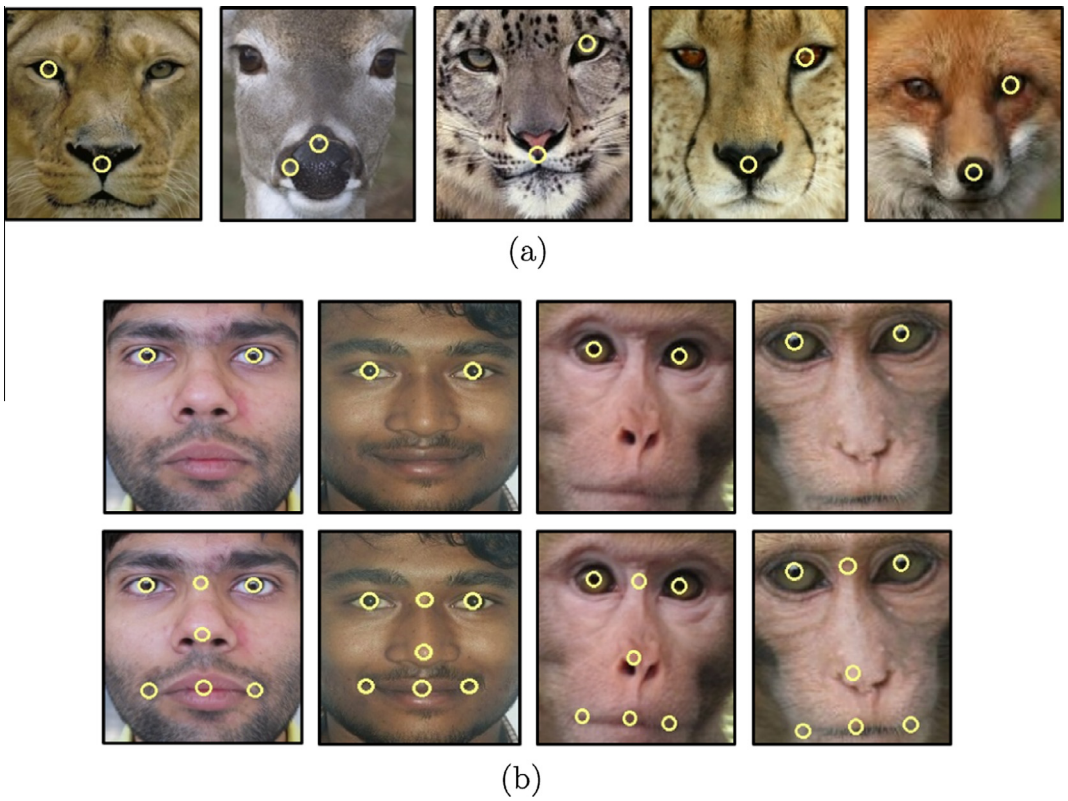

Fig. 24. (a) Results of eye detection on animal face except rhesus monkey and (b) results of eyes and other five control points detection for human and detected target face images.

images are human, cheetah, lion, deer, red fox, snow leopard and rhesus monkey. Animals having images with lower interclass variation are not included in the training set, e.g. leopard and jaguar. Face images of these animals are very similar, which creates difficulty in differentiating the histogram models. Frontal face images were chosen because the dominant facial regions such as eyes, nose and lips can be accurately detected. As a pre-processing part, each image is cropped manually to remove a non facial portion. Our human face database has total 460 distinct images out of which 7 images as shown in Fig. 18(a) are used for training and rest 453 images are used for testing. Here, 63 images are from Indian Face Database [39], 114 images are from CVL face database [41,42] and rest 276 images are from Muct face database [43], respectively. Sample images from each of these database are shown in Fig. 17. The training set consists of 7 images from each different classes viz. human and six different animals which is shown in Fig. 18(a). The target database consists of total 72 distinct images as shown in Fig. 18(b). Authors in [3,4] achieved better classification accuracy using 7 images for training in a class. So following their footsteps, we also use 7 images per class for training.

6.2. Results of target detection

Our method has been tested for each of the 453 distinct human face images with 72 different animal faces. The joint histogram model is generated for each of these image. Then, the *chi-square distance* is calculated between histogram models of each human face with all 72 animal faces. Thus total number of chi square distances are 32,616 (72×453). Fig. 19 shows an example of source and six different target images with their histogram models with five textons per class i.e. $K = 5$. Here $N_{tex} = 35$ (5×7) and $T_{img} = 7$. Table 2 shows the *chi-square distance* between source and these target images as shown in Fig. 19 by considering three different scenarios. These include: similar dimensions source and target images, rotation of target image by 45° and down-sampling of target image by a factor of two. One can observe that, the distance variations are minor for rotation and scale change. Hence, our algorithm takes care of variations in illumination, rotation as well as scale change in source and target images.

In order to test the retrieval accuracy of the proposed approach, we compare our results with JCBIR Content Based Image Retrieval System developed at MIT US [6]. For testing, we include 72 animal face images as target images and set the parameter $K = 5$ and 8 (number of clusters in one class). Each of the 453 human faces are used as query images to the JCBIR. The output sequence of animal faces are then ranked. From the 72 target images the best image is found for each of the 453 source images. The same procedure is repeated to detect the target using the proposed approach. The best target image detected using JCBIR and the proposed approach is same as shown in Fig. 21 but they have different retrieval accuracy. To compare the retrieval accuracy, following criteria is used: The detected image is considered as a false match, if it is not the best target image. Example of false matching is shown in Fig. 20. Let, the number of perfectly matched images and total images are A and B respectively, then matching accuracy in $(\%) = \frac{A}{B} \times 100$. Table 3 lists the comparison of matching accuracy between JCBIR and proposed approach with different values of K. Results show that, the accuracy of the proposed approach is higher when compared to the JCBIR.

In order to test the performance of our target detection, we also compare our approach with two of the recently proposed feature based approaches. These approaches are based on Local Binary Pattern (LBP) and they use Local Ternary Pattern (LTP) [8] and Local Tetra Pattern (LTrP) [9] as features (descriptors). In LBP based approaches, the histogram model is generated using image features only. Therefore, the features follow characteristics of the

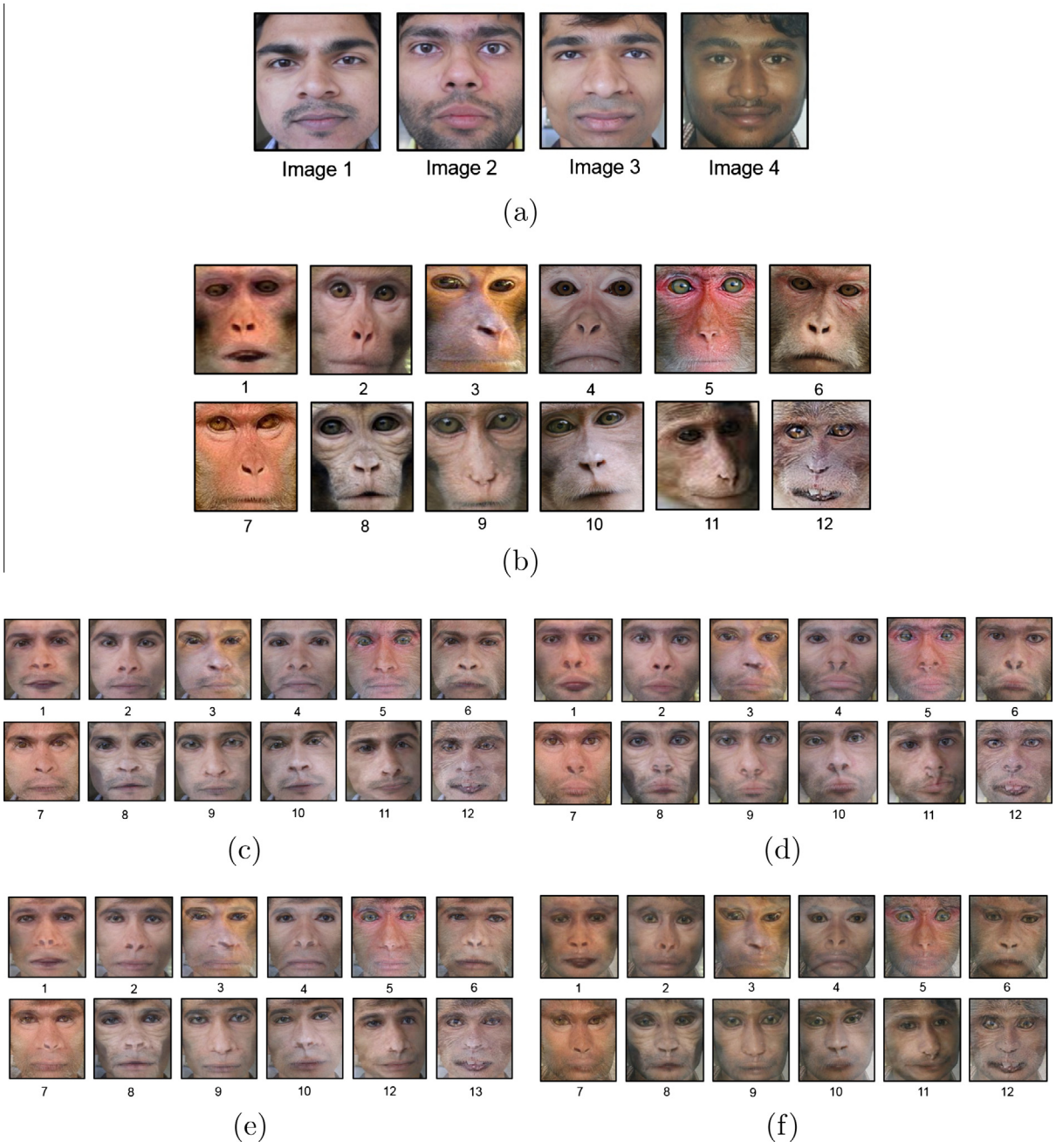

Fig. 25. (a) Four human face images as source, (b) twelve target images of rhesus monkey. Intermediate morphed images for (c) image 1, (d) image 2, (e) image 3 and (f) image 4.

image and not the class to which it belongs. This is the major disadvantage of LBP based approaches. From retrieval perspective it is better if features capture the characteristics of an image as well as class. Therefore, the proposed approach generates the model by learning features from an image as well as from the training set.

We also compare our proposal with two other texton based approaches. They include Maximum Response 8 (MR8) [3] and Markov Random Field (MRF) based approaches [4]. In MR8 based approach, textons are generated using Maximum Response filter bank. The MR8 based approach generates rotation invariant image model but, it requires large filter bank. Also due to blurring, finer local details are lost [3]. The problems of filter based approaches are solved by Markov Random Field (MRF) based approach [4] where textons are generated from training images without filters. However, the MRF based approach for model generation is rotation variant. Here the authors [4] have discussed various solutions to overcome rotation variation and minimize the misclassification error. However, none of these solutions make the model as fully rotation invariant. That is, for a given input image and its rotated version, the method should generate the same output model. By learning rotation invariant features from an image as well as from the training set we build rotation invariant image model.

Table 4 shows the comparison of matching accuracy for different feature based approaches [3,4,8,9]. One can see that, the proposed approach gives better matching accuracy when compared to other feature based approaches (LTP, LTrP, MR8 and MRF). Table 5 shows the comparison of matching accuracy for target detection between MRF based and the proposed approach where

target images are compared with source images for three different scenarios. The scenarios are: (1) without any rotation; (2) target images are rotated by an angle of 90° and (3) target images are rotated by an angle of 45°. Note that, for 0° and 90° of rotation, our approach has the same matching accuracy whereas for 45° the accuracy is reduced due to the effect of interpolation. We would like to mention here that entire set of 453 target images are considered as a single set for comparison. Here, number of textons per class are taken as 5 i.e. $K = 5$. Fig. 22 shows the comparison of matching accuracy with three texton based approaches with three different size of neighborhood i.e. $3 \times 3, 5 \times 5$ and 7×7. Here also all the 453 images are considered as a single set with $K = 5$. Fig. 22 indicates that, higher matching accuracy is achieved by increasing the size of neighborhood. However, it also increases the size of textons which results in increase of time required for texton–vector comparison. The results shown in Tables 3 and 4 indicate that when compared to other approaches, the proposed approach provides better matching accuracy for automatic target detection.

6.3. Results of control points detection

Fig. 23 shows the eye region extraction for each class of facial images. One can note the similarity in regions extracted for human and rhesus monkey (Fig. 23(a) and (b)). Fig. 24(a) is the output of eye detection for animal face images excluding rhesus monkey and Fig. 24(b) includes the output with all seven control points detected for human and rhesus monkey images. The small circles shown in the images indicate the detected control points. This clearly indicates that, for the chosen target images, rhesus monkey matching with the source image. One may include different class of images in training set and based on that our algorithm finds the suitable target image for a given source. Note that, for the best matched animal image i.e. rhesus-monkey, the control points for source and destination features match very well. Once these control points are detected, morphing between these images is done using the approach proposed in [2].

6.4. Morphing and perceptual measure

The effectiveness of automatically selected target image for morphing is validated through following experiment. Four images are randomly selected as source images from human face database and each of twelve images of rhesus monkey are selected as target images. The source and target images are shown in Fig. 25(a) and (b) respectively. As per the method discussed in Sections 3 and 4, seven control points are detected and morphing is carried out between each source image and all twelve target images. Intermediate morphed images using image 1, image 2, image 3 and image 4 are shown in Fig. 25(c)–(f) respectively. The goodness of these results is validated using subjective testing by humans. This is done as follows: four source images of Fig. 25(a) and their respective morphed images of Fig. 25(c) and (d) are given as an input to Amazon Mechanical Turk [10]. We then query their human

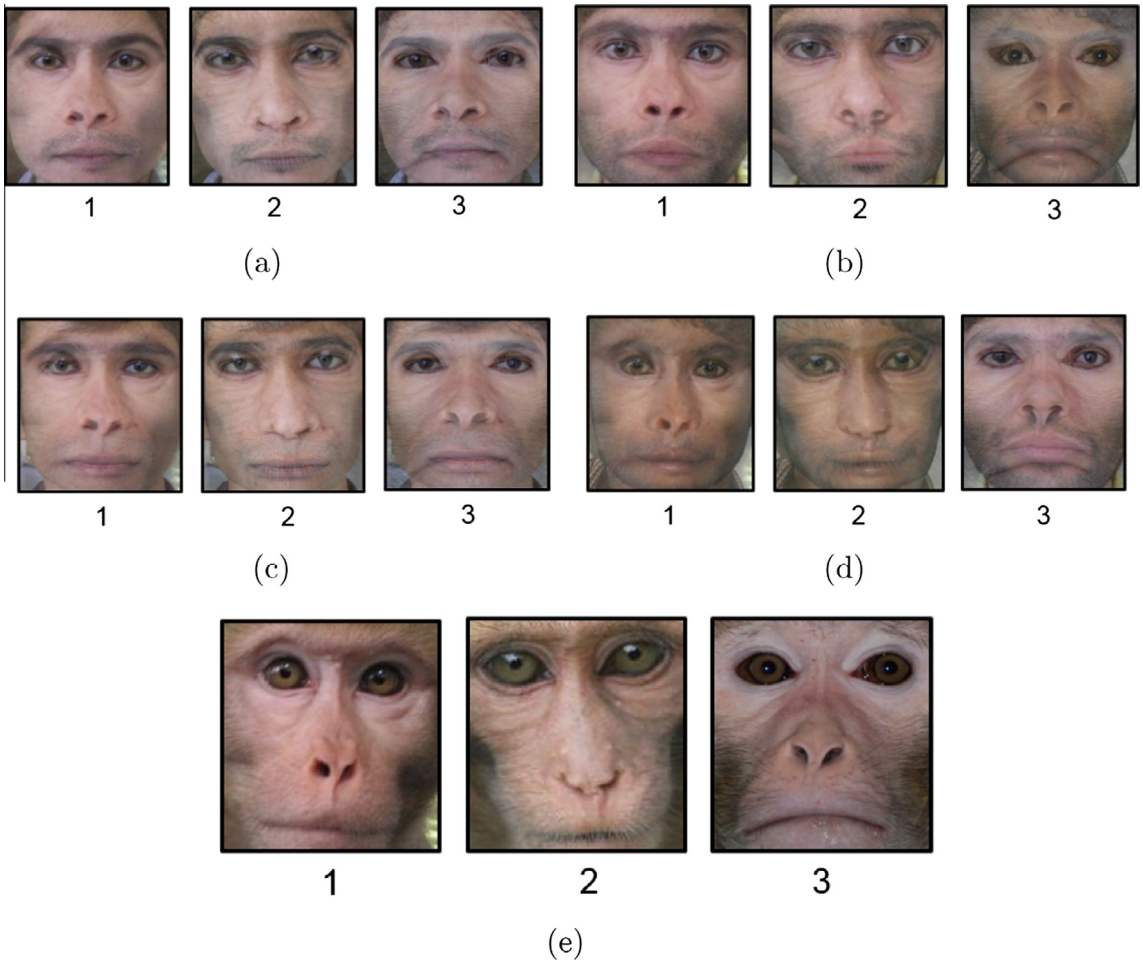

Fig. 26. Three best morphed images selected by human workers of Amazon Mechanical Turk [10] using (a) image 1, (b) image 2, (c) image 3, (d) image 4 as a source of Fig. 25(a) and (e) their respective target images.

workers with the following question: "Which of these intermediate morphed images possess features alike source image?" and 1000 responses are registered. For each of the source image, best three morphed images as selected by the workers are shown in Fig. 26(a) and (b), respectively. The corresponding target images are shown in Fig. 26(c). 37% of the Amazon Mechanical Turk workers liked morphed image 1 of Fig. 26(a)–(d) while 22% and 17% of the workers favored image 2 and 3 respectively. 24% of the workers opinion was spread across remaining nine morphed images. One can observe that, the first target image corresponding to the three best morphed results selected by the workers of Amazon Mechanical Turk is same as the target image selected by our approach. Thus, human annotation validates efficacy of our proposed approach while selecting target image automatically. Note that, one can increase the number of inputs to Amazon Mechanical Turk for more effective results. However, simultaneous assessment of large data may confuse the human workers, which may result in lowering the accuracy.

7. Conclusion

Proposed approach shows that, for a given human face and set of target images, rhesus monkey face is best suited for morphing. Method detects target image without human intervention. Moreover, region of interest and control points are also automatically detected. The histogram model is robust to rotation, and combining it with variable length binning provides in-variance against local contrast changes and partial resistance to scale variations. These changes resulted into a higher matching accuracy of the target image using the proposed approach in comparison to JCBIR and other feature based approaches. More importantly the same approach is used to detect the control points in a source as well as in the target image for morphing. Quantitative detection results are verified with subjective human annotations using Amazon Mechanical Turk. These outcomes validate the detection results of our proposal. It indicates that auto detected animal face image maps well to the source image, resulting in a better morphing process.

References

[1] G. Wolberg, Recent advances in image morphing, Comput. Graph. Int. (1996) 64–71
[2] G. Wolberg, Image morphing: a survey, Visual Comput. 14 (8) (1998) 360–372.
[3] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, Int. J. Comput. Vision: Spec. Issue Texture Anal. Synth. 62 (1–2) (2005) 61–81.
[4] M. Varma, A. Zisserman, A statistical approach to material classification using image patch exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2032–2047.
[5] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vision 43 (1) (2001) 29–44.
[6] JCBIR. <https://code.google.com/p/jcbir/>, 2010.
[7] Y. Latha, B. Jinaga, V. Reddy, Content based color image retrival via wavelet transform, Int. J. Comput. Sci. Netw. Secur. (2007) 38–45
[8] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Trans. Image Process. 19 (6) (2010) 1635–1650.
[9] M. Subrahmanyam, R.P. Maheshwari, R. Balasubramanian, Local tetra patterns: a new feature descriptor for content-based image retrieval, IEEE Trans. Image Process. 21 (5) (2012) 2874–2886.
[10] Amazon Mechanical Turk. <https://www.mturk.com/mturk/welcome>.
[11] G. Wolberg, Skeleton-based image warping, Visual Comput. 5 (1 & 2) (1989) 95–108.
[12] T. Beier, S. Neely, Feature-based image metamorphosis, in: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 1992, pp. 35–42.
[13] S. Lee, G. Woberg, K.-Y. Chwa, S.Y. Shin, Image metamorphosis with scattered feature constraints, IEEE Trans. Visual Comput. Graph. 2 (4) (1996) 337–354.
[14] S. Lee, G. Wolberg, S.Y. Shin, Polymorph: morphing among multiple images, IEEE Comput. Graph.Appl. 18 (1) (1998) 58–71.
[15] A.W.F. Lee, D. Dobkin, W. Sweldens, P. Schrder, Multiresolution mesh morphing, in: Proceedings of SIGGRAPH, 1999, pp. 343–350.
[16] A. Lee, D. Dobkin, W. Sweldens, P. Schrder, Multiresolution mesh morphing, in: Proceedings of SIGGRAPH, 1999, pp. 343–350.
[17] S.M. Seitz, K.N. Kutulakos, Plenoptic image editing, in: Proceedings of the 6th International Conference on Computer Vision, 1998, pp. 17–24.
[18] L. Wang, S. Lin, S. Lee, B. Guo, H.-Y. Shum, Light field morphing using 2D features, IEEE Trans. Visual. Comput. Graph. 11 (1) (2005) 25–34.
[19] Y. Luo, M.L. Gavrilova, P.S.P. Wang, Facial Metamorphosis using geometrical methods for biometric applications, Int. J. Pattern Recogn. Artif. Intell. 22 (2008) 555–584.
[20] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH, 1999, pp. 187–194.
[21] F.H. Pighin, R. Szeliski, D. Salesin, Resynthesizing facial animation through 3D model-based tracking, in: International Conference on Computer Vision, 1999, pp. 143–150.
[22] M. Bichsel, Automatic interpolation and recognition of face images by morphing, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, 1996, pp. 128–135.
[23] C. Zhang, F.S. Cohen, 3-D face structure extraction and recognition from images using 3-D morphing and distance mapping, IEEE Trans. Image Process. 11 (11) (2002) 1249–1259.
[24] S.-Y. Lee, K.-Y. Chwa, S.Y. Shin, Image metamorphosis using snakes and free-form deformations, in: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 1995, pp. 439–448.
[25] V. Zanella, G. Ramirez, H. Vargas, L.V. Rosas, Automatic morphing of face images, in: Proceedings of the 9th International Conference on Adaptive and Natural Computing Algorithms, 2009, pp. 600–608.
[26] T. Ojala, M. Pietikäinen, Unsupervised texture segmentation using feature distributions, Pattern Recogn. 32 (3) (1999) 477–486.
[27] M. Varma, A. Zisserman, Unifying statistical texture classification frameworks, Image Vis. Comput. 22 (14) (2004) 1175–1183.
[28] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[29] G.A. Papakostas, D.E. Koulouriotis, E.G. Karakasis, V.D. Tourassis, Moment-based local binary patterns: a novel descriptor for invariant pattern recognition applications, Neurocomputing 99 (2013) 358–371.
[30] S. Baskan, M.M. Bulut, V. Atalay, Projection based method for segmentation of human face and its evaluation, Pattern Recogn. Lett. 23 (14) (2002) 1623–1629.
[31] T. Hamada, K. Kato, K. Kawakami, Extracting facial features as in infants, Pattern Recogn. Lett. 21 (5) (2000) 407–412.
[32] S. Asteriadis, N. Nikolaidis, I. Pitas, Facial feature detection using distance vector fields, Pattern Recogn. 42 (7) (2009) 1388–1398.
[33] Z. ming Qian, D. Xu, Automatic eye detection using intensity filtering and K-means clustering, Pattern Recogn. Lett. 31 (12) (2010) 1633–1640.
[34] M.B. Yilmaz, H. Erdogan, M. Unel, Facial feature extraction using a probabilistic approach, Signal Process.: Image Commun. 27 (6) (2012) 678–693.
[35] R.-L. Hsu, M. Abdel-Mottaleb, A.K. Jain, Face detection in color images, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 696–706.
[36] F.Y. Shih, C.-F. Chuang, Automatic extraction of head and face boundaries and facial features, Inf. Sci. 158 (1) (2004) 117–130.
[37] D. Ruprecht, H. Müller, Image warping with scattered data interpolation, IEEE Comput. Graph. Appl. 15 (2) (1995) 37–43.
[38] M. Bichsel, Automatic interpolation and recognition of face images by morphing, in: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition, 1996, pp. 128–139.
[39] V. Jain, A. Mukherjee, The Indian Face Database. <http://vis-www.cs.umass.edu/vidit/IndianFaceDatabase/>, 2002.
[40] Google Images. <http://images.google.com/>, 2013.
[41] P. Peer, CVL Face Database. <http://www.lrv.fri.uni-lj.si/facedb.html>, 2010.
[42] F. Solina, P. Peer, B. Batagelj, S. Juvan, J. Kovac, Color-based face detection in the '15 seconds of fame' art installation, in: Conference on Computer Vision Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects, 2003, pp. 38–47.
[43] S. Milborrow, J. Morkel, F. Nicolls, The MUCT Landmarked Face Database, Pattern Recognition Association of South Africa. <http://www.milbo.org/muct>.