Received: January 8, 2019

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Position Invariant Spline Curve Based Image Retrieval Using Control Points

Sandeep Dwarkanath Pande[1]* Manna Sheela Rani Chetty[1]

[1]*Department of Computer Science and Engineering, KLEF Vaddeswaram,*
Guntur District, Andhra Pradesh, India
* Corresponding author's Email: sandeep7887pande@gmail.com

Abstract: In this paper Cubic Bezier curve-based image retrieval system is proposed. This system evaluates similarity of each image in its database to a query image in terms of shape characteristics. Then, returns those images within a desired range of similarity. The proposed system determines nonlinear relationship between image's features for more accurate similarity comparison between query image and existing database images. Among existing approaches to shape feature analysis, statistical approach to extract shape features is adopted here. It works in form of control points of spline curves from both given query image and images of available database. These control points are further used to find out the Fourier Descriptors. Control points and Fourier descriptors are used for image retrieval in proposed system. With the vast number of images available on-line, quality CBIR systems are critical. The performance and results obtained by proposed system are compared to other CBIR systems. Comparison reveals proposed system performance is state of the art. In many parameters it outperforms other CBIR systems.

Keywords: CBIR, Shape, Fourier descriptor, Retrieval, Spline curve, Bezier curve, Control points.

1. Introduction

Due to the advancement in multimedia, social media digital devices mobile applications and discrete-level furtherance in digital world in recent years enormous amount of visual information is generated rapidly. This rapid growth in the number of large-scale repositories has brought the need for an image retrieval system. An effective and efficient image retrieval system is always crucial to retrieve images of user's interest based on query from enormous amount of image database. Image retrieval is also substantial to access visual information and generate knowledge from such information. Text Based Image Retrieval (TBIR) is one of the orthodox approaches that retrieve images based on text input. Here images are annotated with text. A text query or keyword was entered and the images with similar text annotations were retrieved. This approach is limited by manual annotations which is costly and time consuming. They also fail to deal with discrepancy of subjective perception. Some media contents are difficult to describe concretely in words. This promotes for introduction of new image retrieval system which is based on visual contents of the image known as Content Based Image Retrieval (CBIR). Fig. 1 represents general architecture of CBIR System [1]. CBIR System has a database of images. The system retrieves the images from this database on the basis of input which can be specified by several ways. The input is basically, content of the image i.e. either a texture or shape present in the image. We can even specify the input as a combination of these image features. The state of the art in the CBIR systems is to search images in database that are "close" to the query image using some similarity measure. The current CBIR systems capture image features that represent properties such as color, texture, shape, or text annotations of the objects in the query image and try to retrieve images from the database with similar features.

In this paper the constraint of accurate feature extraction is focused and a new precise approach for modified 2 stage approximation method to extract control points of a cubic Bezier curve for high variant feature description is proposed. Extraction of accurate control points from a Bezier curve with an efficient method is also a big challenge. The modified 2 stage algorithm requires the control points of the Bezier curve in a sorted order of tracing path of the curve. The proposed method considers the Bezier curve points in polar coordinate form which reduces the overhead of arranging Bezier points in traced order. The proposed method works better for both concave and convex Beziers. In this paper the constraint of optimal feature representation is further focused and a new feature vector Fourier Descriptor (abbreviated as FD henceforth) to represent the primary features (control points) is proposed. The proposed method achieves feature selection accuracy. The approach signifies an advantage in low resource overhead in computing shape feature using Bezier curves. The proposed method stores the feature vector as Fourier Descriptors which are very less in count (20 values per image) which results in very less storage requirement and reduced computation in feature matching which results in increased performance as compared to other methods. The proposed method also yields accurate results irrespective of position and orientation of the object in image, making the approach position and rotation invariant.

The rest of the paper is organized as follow. A Brief review of the recent researches related to CBIR is given in Section 2. In section 3, the pre-processing steps required to extract the boundary is presented. Bezier curves, their properties and the proposed approach for control point extraction as a primary shape feature is described in section 4. Section 5 describes representation of shape signatures into FD as an optimal shape feature. In section 6, experimental results are reported to show the performance of the proposed approach. Concluding remarks are offered in section 7.

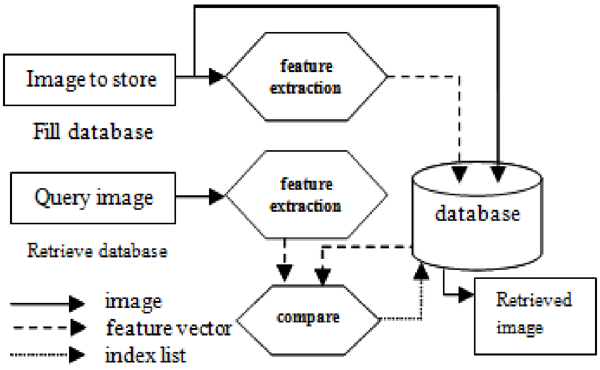


Figure. 1 General architecture of CBIR-system

2. Related work

In [2] to derive edge features, a combination of invariant moments and edge direction histogram is proposed. In various approaches, moments are used as a shape descriptor to define the shape feature. In [3] an angular radial transform (ART) descriptor, for MPEG-7 region-based shape description is proposed. The method is defined as a variation of Independent component analysis of shape descriptor using Zernike shape moment descriptor. Wherein edge based approaches are the simplest mode of shape representation, in most of the image representation, edge operators derive coefficients out of the bounding regions. This extra information results in computational overhead, making the system slower in process. To derive more precise shape description, contour-based coding was developed. A contour-based learning approach is defined in [4]. The approach of contour is a bound region growing method where the outer bounding region is extracted via a region growing approach to derive image representation. A binary contour-based similarity measure is developed in [5]. This approach, defines a new similarity measure for images with weighted distance measure for contour using string match approach. The methods of the contour-based are mainly polygonal approximation, wavelet descriptors, or scale space [6, 7], wherein the region-based methods are mainly geometric moment invariants, and orthogonal moments [8]. In addition to the edge and contour-based coding various other approaches such as in [9] a shape feature representation using shape context is presented. The approach defined a histogram-based representation of bounding coefficient describing the relative relation of edge regions in the image. An approach of medical image segmentation using active contour approach is outlined in [10]. The approach developed a segmentation method for the synovial region in an ultrasound image. The approach of contour evolution and segmentation using active contour model and SVM classifier model is developed in learning of ultra sound images. A similar contour-based segmentation using a fine-grained approach for real time object detection in content-based multimedia application is developed in [11]. The object detection for multi category objects detection using a confidence score is developed. A dynamic region bound region of interest (ROI) detection is proposed for the object detection. However, the contour based or the other techniques such as graph based, context based etc. defines the overall bounding contour, wherein the variations in the feature coefficients are very large.

Each projection in the contour region is taken as a feature, which leads to large feature data set. To overcome the problem of large feature vectors, curvature coding has emerged in recent past. A curvature-based scale space representation is presented in [12]. For the environmental study of the plant structuring, leaf image detection using shape-based detection is presented in [13]. The approach defines a segmentation method for the detection of leaf image based on the shape difference and principal component analysis (PCA). Bezier curves [14] are also used as a feature representative curve in image recognition. The method operates on fitting of Bezier curve on the boundary of edge region of an image, and extracting control point representation for this curve. The dominant curvature peaks are used in the control point extraction which is used for feature description. The feature vector size is larger in this approach resulting more comparison time to retrieve images. A quaternionic–Bezier curve in three-dimensional space projections for interpolation is proposed in [15]. Here a five-point interpolation or a three point, 2 tangent projections is outlined. The 3D interpolation is used in the image representation where control points are derived using the Bezier curve representation. In [16] Bezier curve-based Sea Surface Distribution Modeling method is proposed for both of these methods are having less computation speed as these methods involve solution of system of linear equations to compute the control points. A contour-based leaf classification is presented in [17] that estimate a quadratic curve from 3 landmark points: apex, centroid and end of petiole. Then it aligns the leaf contour projection on quadratics curve. Then it normalizes mid axis of leaf. By using mid leaf axis and leaf contour it identifies the tangent’s direction of leaf contour which is used as a feature. The approach is having many trigonometric and curve approximation computations involved to reach to the feature vector which reduces performance of the system.

An effective representation method was proposed in [18] that makes use of both concavities and convexities of all curvature points. The multi scale convexity concavity (MCC) defines the feature for different Gaussian kernels and derived as the boundary point of each contour point of the previous level. The approach of curvature coding, results in lower feature descriptors for image retrieval. However, in such coding, features are extracted based on a threshold of the curvature plot, and values with higher magnitude are selected. This approach of feature selection process discards the lower variation information considering as noise. However, in various image samples curvature with variations existing for a lower time period exist. So, this assumption of feature selection process minimizes the descriptive features relevancy w. r. t. image representation. In the recognition of control point where images have high varying short period variations, the feature points are large in count. This feature description results in large memory storage and higher search overhead. The larger descriptive feature also leads to higher probable misclassification in recognition.

3. Pre-processing

The shapes considered in this paper are outline shapes which can be described as single plane closed or open curves. The shapes in our database are obtained either from silhouette real world objects or from user-drawn shapes; the shapes are in the form of gray level images. The pre-processing is to extract the boundary information, or coordinates of the boundary, from the input image. The diagram for pre-processing is shown in Fig. 2. The first step in the pre-processing is to binarize the shape image; a simple thresholding is applied to convert a gray level shape image into binary image.

In reality, shape images are often corrupted with noise, as a result, shape obtained after thresholding usually has noise around the shape boundary. Therefore, a denoising process is applied [19]. The denoising process eliminates those isolated pixels and those isolated small regions or segments. For a non-silhouette shape, the shape boundary is not always connected. Thus, an m-connectivity connection technique is used to fill the gaps between boundary points [20]. The shape is then traced using an 8-connectivity contour tracing technique to obtain the shape boundary coordinates.

Some examples of pre-processing are shown in Fig. 3. Sometimes bounded regions are not readily available, for such shapes boundary is detected. Then m-connectivity is applied for the internal parts of the curve and thinning is applied to get the uniform curve [21].

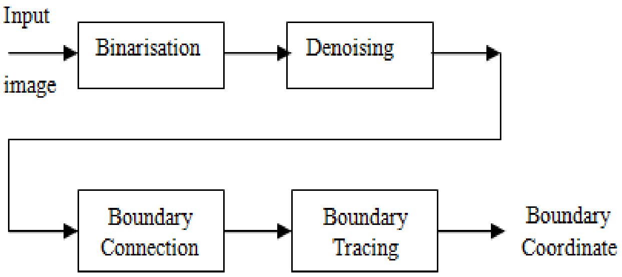


Figure. 2 Pre-processing of shape image

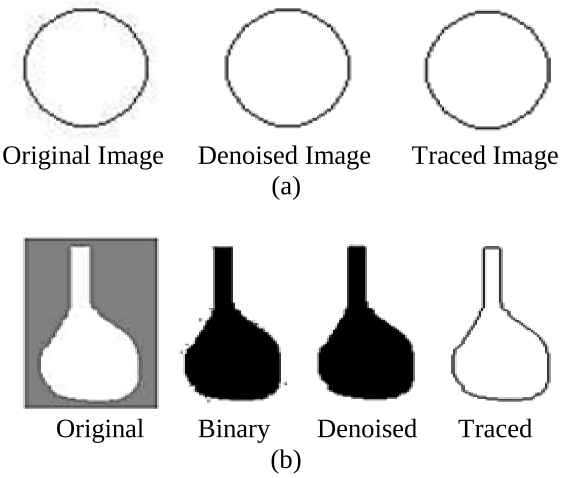


Figure. 3 Examples of pre-processing: (a) user drawn shapes and (b) silhouette shape

Object reorganization system based on global features matching such as area, size, moments are very robust with regard to noise, but their performance decreases in case of images with partially overlapping objects [22]. Overlapping objects pose a serious challenge for model-based detection task. The following part of this section discusses overall pre-processing of given image. Also, to get different shapes of objects present in the images using boundary detection and thinning algorithms. The shapes which are considered for further processing are irregular shapes made up of splines.

A. Query by Shape Content:

System accepts input in two ways.

1. Input by Existing Image: We are directly extracting the spline curve features from the query image as described later in this paper.
2. Input by Drawing Shape: With drawing tools Here we provide CBIR paint tool to the user, that tool include pencil and eraser. Once image is drawn by user that image is saved at specified location and user can select same image as query image.

B. Pre-processing

In order to get dominant points along the boundary, boundary is differentiated from rest of the object. For this purpose, following steps are performed on given image.

1) Plotting Histogram

Histogram of an image represents the relative frequency of occurrence of various gray levels in the image [23]. The plot obtained between gray levels v/s their frequency is known as gray level Histogram. From Histogram, approximate level of threshold can be extracted.

2) Threshold of an image

Threshold plays important role to detect the boundary of an object. Threshold is one of the simplest methods of segmentation [23]. In this, brightness value of each pixel is compared to threshold value and the pixel is assigned to either background or object.

3) Boundary detection

Boundary detection or contouring is used to extract shapes from given image. Authors of work [24] have discussed contouring or boundary detection algorithms in details. In this work, Marching squares algorithm with its latest variations as discussed in [24, 25] is used for boundary detection. This algorithm works successfully for RGB as well as grayscale images.

4) Thinning

Purpose of thinning is Convert binary shapes obtained from boundary detection or threshold to 1-pixel wide lines [21].

After thinning of shape boundary, control points are identified. In addition to these points, points where curve has maximum variations are also selected as dominant points. As every shape is made up of curved boundaries and such a shape can be represented/defined by a Bezier spline curve. Our approach uses an approximation method to identify the control points of Bezier.

4. Shape feature extraction

A. Bezier curves:

Bezier spline curves have a number of properties that make them highly smooth, useful and convenient for curve and surface drawing. They are also easy to implement [26]. Due to these reasons, Bezier curves are very popular and are widely used in design. Various Graphics packages also contain tools to use such curves. Generally, a Bezier curve can be fitted to any number of control points. Each curve is associated with a polynomial equation which defines the shape of a curve. The polynomial equation has degree one less than the total number of control points. If it has 4 control points then such curve is known as cubic Bezier curve. The following Fig. 4 shows Bezier curves with 3 and 4 control points. The Bezier curves can be specified with blending functions.

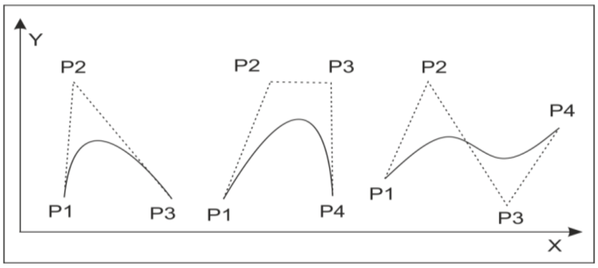


Figure. 4 Bezier curves with 3 and 4 control points

The curves can be defined mathematically in three ways, implicitly, explicitly and parametrically as (i) $f(x, y) = 0$, (ii) $y = h(x)$, and (iii) $x = f(t)$, $y = g(t)$ respectively, where $f(x, y)$, $h(x)$, $f(t)$, $g(t)$ are polynomials. It is possible to get the implicit equation from the parametric one and find the parameter t so that $(x, y) = (f(t), g(t)) = (a, b)$. A Bezier curve $Q(t) = [Q_x(t), Q_y(t)]$ of degree n can be defined in terms of $n + 1$ control-point points: $P_i = (x_i, y_i)$, with i varying from 0 to n. These coordinate points can be blended to produce $Q(t)$, which describes the path of a Bezier curve between P_0 and P_n.

$$Q(t) = \sum_{i=0}^{n} P_i \, B_{i,n}(t) \qquad (1)$$

The Bezier blending functions $B_{i,n}(t)$ are the Bernstein polynomials and are defined as:

$$B_{i,n}(t) = C(n,i)t^i(1-t)^{n-i}, 0 \leq t \leq 1 \qquad (2)$$

Where the $C(n, i)$ are the binomial coefficients:

$$C(n,i) = \frac{n!}{i!(n-i)!} \qquad (3)$$

The Bezier curves have various important features & Properties which makes them popular in design [26]. Out of them following are crucial for proposed method:

1. Bezier curve can be specified with boundary conditions or characterizing matrix, or with blending functions
2. It passes through first and last control point.
3. The slope at the beginning of the curve is along the line joining the first two control points, and the slope at the end of the curve is along the line joining the last two endpoints.

In addition to this, position & movement of the control points influences the change in the shape of the curves. The control points control the shape of the portion of curve which is near to it and each seems to exert a pull on the portion of the curve near to it. The Bezier curves can be evaluated based on these properties [27].

Let P_0 and P_n be the two end points, P_i $(i \in [1, n - 1])$ be the inner control points of the to-be-derived Bezier curve Q $(t = k / N)$, $(k \in [0, N])$, and $\hat{t}_j$ $(j \in [1, 2])$ be the unit tangent vectors at the end points of Bezier curve. Then, find a Bezier curve $Q(t)$, such that the sum of square distance errors (RMS) from each data point to the curve is minimal. So here approximating control point positions and finding an approximated Bezier curve $Q(t)$ using Eq. (1) is done. That is, find $Q(t)$ to minimize E in the following expression (4).

$$E = \sum\nolimits_{k=0}^{N} |Q(t = k/N) - q(k)| \qquad (4)$$

Where $Q(t = k / N)$ is corresponding curve points of approximated Bezier curve, $q(k)$ is the data point to be fitted with the Bezier curve. For cubic Bezier curve, Eq. (1) can be written as:

$$Q(t) = P_0B_{0,3}(t) + P_1B_{1,3}(t) + P_2B_{2,3}(t) + P_3B_{3,3}(t) \qquad (5)$$

The inner control points of curve $Q(t)$ are expressed by:

$$P_1 = P_0 + \alpha\hat{t}_1$$
$$P_2 = P_3 + \beta\hat{t}_2 \qquad (6)$$

Where α and β are scalars used to specify the locations of the control points P_1 and P_2 on the unit tangent vectors represented by $\hat{t}_1$ and $\hat{t}_2$ respectively. Using Eq. (6); Eq. (5) can be expanded as:

$$Q(t) = P_0\{B_{0,3}(t) + B_{1,3}(t)\} + \alpha\hat{t}_1B_{1,3}(t) + \beta\hat{t}_2B_{2,3}(t) + P_3\{B_{2,3}(t) + B_{3,3}(t)\} \qquad (7)$$

Substituting the expression for $Q(t)$ in Eq. (4), we have:

$$E = \sum_{k=0}^{N} \left| A(t) + \alpha\hat{t}_1B_{1,3}(t) + \beta\hat{t}_2B_{2,3}(t) - q(k) \right| (8)$$

Where,
$A(t) = P_0\{B_{0,3}(t) + B_{1,3}(t)\} + P_3\{B_{2,3}(t) + B_{3,3}(t)\}$.

This is a nonlinear equation which cannot be solved by the traditional least-squares techniques [7]. So, in order to find the control points of cubic Bezier curves, a new approach with modified two-stage learning algorithm is discussed in following subsection.

B. Control points identification of Cubic Bezier curve:

Assuming N is the total number points of the Bezier curve along the shape boundary. Using these curve points, calculate control points of the Cubic Bezier curve. First sort given curve pixels w. r. t. tracing path of given curve. Then apply improved curve height calculation method & improved two stage learning algorithm [28, 29]. Let the four control points to be determined are P_0, P_1, P_2, and P_3, the i^{th} control point of the original Bezier curve be $P_i(x_i, y_i)$. The i^{th} approximate control point be $P_i^{'}(x_i^{'}, y_i^{'})$ for i = 0, 1, 2, 3, . . . , N respectively.

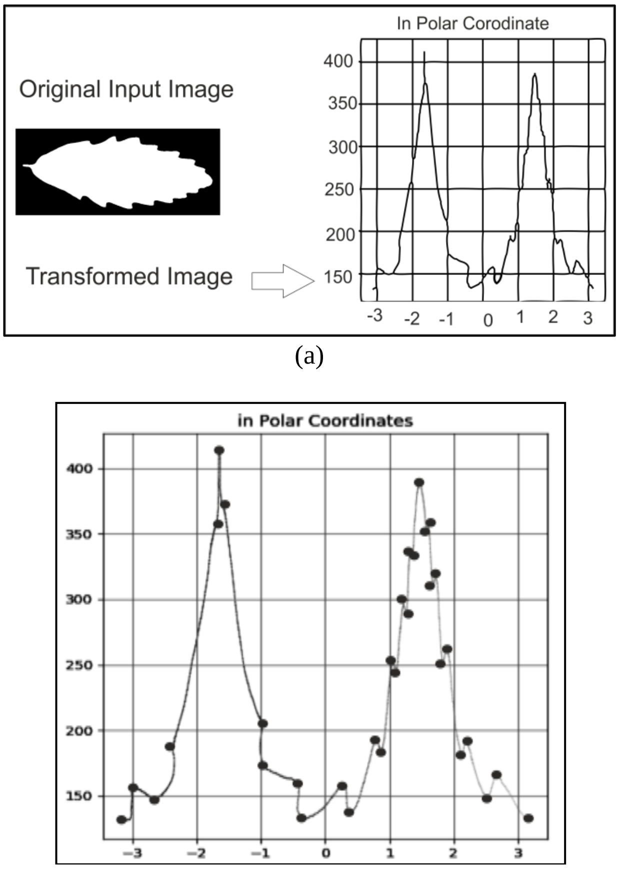
1) 1st Stage:

This stage is dedicated for positioning the control points at approximate positions. The beginning positions of the control points are determined as follows:

a) Modified procedure for finding the 1st and 4th control points:

As per the property of Cubic Bezier curve, P_0 and P_3 control points are initial and terminating curve points. This requires curve pixels to be in order of tracing path [25, 26]. Any curve points are represented in computer graphics as *pixel (x, y)* form. In this work, ordering pixels is done by converting image from Cartesian coordinate system to Polar coordinate system. For this purpose, image is transformed to match its center with Origin. Polar coordinate representation converts image to a linear curve as shown in Fig. 5 (a). Later, in Polar coordinate system points are ordered using their angle as it is clear from Fig. 5 (a).

This linear curve representation is also called as time series representation of an image. Also, to make this representation position invariant center of image is aligned to origin. Additionally, polar coordinates for all points are normalized. Then, this linear curve is segmented into 10 nearly equal parts. Size of each segment is chosen to be roughly one tenth of the size of complete curve. Also, extreme points of the curve are selected and stored separately. Some selected points on query image are shown in Fig. 5 (b). Final image is stored in dataset as set of selected points, control points and FD for all 10 segments.



(a)

(b)

Figure. 5 Linear curve: (a) transforming image to linear curve and (b) selected points from linear curve

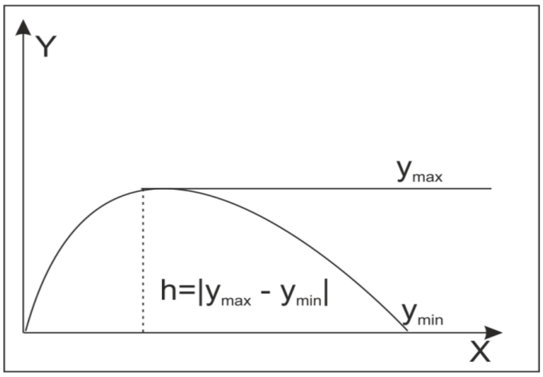


Figure. 6 Calculation of height of the curve

b) Calculation of Height (h) of the curve:

As shown in the Fig. 6, consider start & end points of given curve. If both of them are having same y coordinate value, then *ymin* & *ymax* can be computed from pixel values. Also, here absolute difference between *ymin* & *ymax* is height (h) of the curve.

Now next case is when start and end points of the curve are not having same y coordinate. So, in this case first find out slope of line joining start and end points. The rotation angle θ is tan inverse of this slope as shown in Fig. 7. Now rotation by θ is

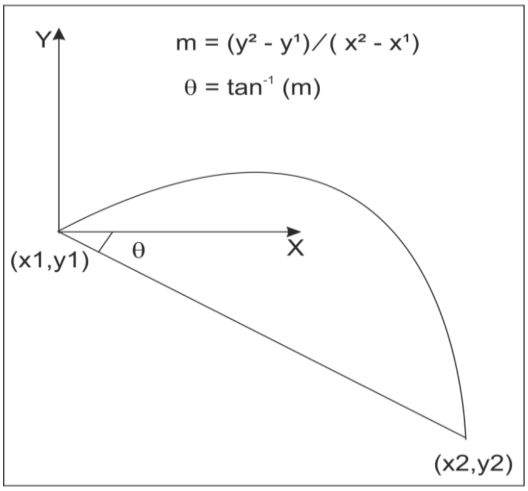


Figure. 7 Slope and rotation angle θ calculation

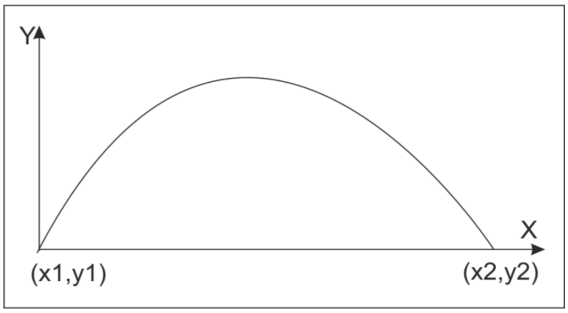


Figure. 8 Resulting curve after applying rotation by an angle θ in an anti-clockwise direction

applied to curve points about first control point in anti-clockwise direction. This brings first and last control point at same y position as shown in Fig. 8 [30]. The Eq. (9) can be used to achieve this.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & r_x(1-cos\theta)+r_y sin\theta \\ sin\theta & cos\theta & r_y(1-cos\theta)-r_x sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{9}$$

Where *(r_x, r_y)* is the point about which x, y is to be rotated by an angle θ.

Now height (*h*) calculation is same as discussed earlier. Hence our last control point is temporarily present at rotated position by an angle θ.

a) Assumption of 2nd and 3rd control points:

Firstly, positions of second and third control points, are approximated on the tangents to the given curve. These tangents are at beginning and end parts. Control points lie at vertical distance of *H* from the base line [31, 32]. Here, *H* may take one of the 3*h*/4, *h*, 4*h*/3, and 1.6*h* values. From practical observations these values of *H* get better results [32]. Fig. 9 shows assumed control points at distances 3*h*/4, 4*h*/3, *h* and 1.6*h* for a sample curve. The reason behind assuming control pints on the tangent lines is the property 2 of the Bezier curve as listed in previous section.

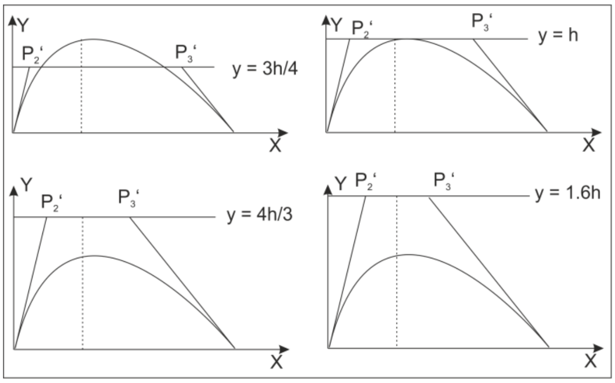


Figure. 9 Assumed control points positions at distances different distances from base line

To work out the assumed control point positions firstly draw a line that is parallel to the line joining initial and terminating point at a distance either 3*h*/4, *h*, 4*h*/3 and 1.6*h*. Then draw two tangents of the curve at the beginning and terminating part of the curve respectively. The tangent can be easily drawn at the beginning part by finding out the slope of line joining 1st & 10th point & extending this line up to *H*. The tangent at the terminating part can be easily drawn by finding out the slope of line joining *n*th & *n* - 10th point & extending such a line up to H. Then the intersection points between parallel line and tangents are computed [33] and these are the initially assumed 2nd and 3rd control points. Here all the assumed control points at different distance are shown in Fig. 9.

Let *($x\_assumed_2$, $y\_assumed_2$), ($x\_assumed_3$, $y\_assumed_3$), (x_1, y_1), (x_4, y_4)* be the assumed control point 2, assumed control point 3, Control Point 1 and Control Point 4 respectively. Hence initial position of assumed control points can be found by following Eq. (10):

$$\begin{gathered} x\_assumed_2 = x_1 + H/m_1 \\ y\_assumed_2 = y_1 + H \\ x\_assumed_3 = x_4 + H/m_2 \\ y\_assumed_3 = y_4 + H \end{gathered} \tag{10}$$

where m_1 and m_2 are the slopes of lines joining 1st & 10th curve points and nth & n -10th curve points respectively.

1) 2nd Stage:

This stage is dedicated to the determination of the exact locations of the control points. At this

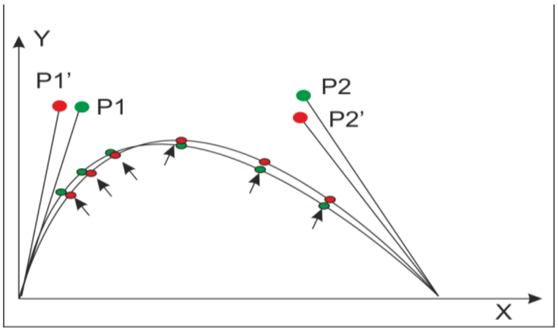


Figure. 10 Calculation of sum of the difference between actual & corresponding points of original & approximated curve

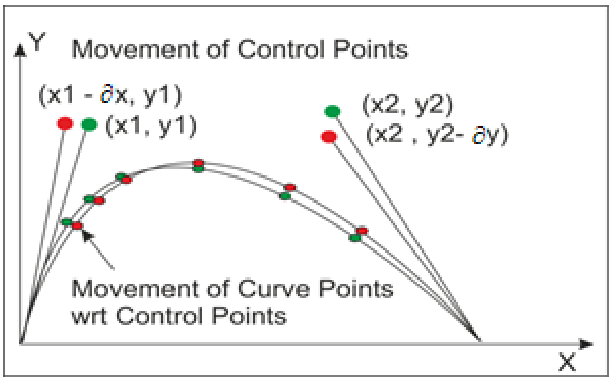


Figure. 11 Movement of approximated control points to minimize the RMS error

stage, finding RMS error E_1 is achieved. E_1 is the summation of the difference between points of given curve $Q(t)$ and points of generated Bezier curve $Q'(t)$. In $Q'(t)$ control points, P_1' and P_2' are assumed as shown in Fig. 10. Here, aim is to minimize the E_1 to get the exact locations of control points. To achieve this, assumed control points are moved initially by a large step. Then RMS error E_1 is calculated. If E_1 is in under permissible value (here, 0.000001) then it is minimum error and control points are finalized. Otherwise assumed control points are moved by half of the current step value, iteratively.

Bezier curves can be obtained by considering a curve in parametric expressions (Eq. (11)) [27]. In these equations' parameter t is between 0 and 1. t value is changed by $1/n$ if the given curve is having n points.

$$X(t) = X_0(1-t)^3 + 3t(1-t)^2X_1 + 3t^2(1-t)X_2 + t^3X_3$$
$$Y(t) = Y_0(1-t)^3 + 3t(1-t)^2Y_1 + 3t^2(1-t)Y_2 + t^3Y_3 \tag{11}$$

Let N+1 be total number of points on given curve. The coordinates of corresponding points are $Q(t) = [Q_x(t), Q_y(t)]$, $Q'(t) = [Q'_x(t), Q'_y(t)]$, $(t=0, 1/N, 2/N, \ . \ . \ . \ , 1)$. Thus, the RMS error E_1 can be expressed as:

$$E_1 \triangleq \sum_{k=0}^{N} |Q(t=k/N) - q(k)|$$
$$= \sum_{k=0}^{N} \sqrt{\begin{matrix} \{Q_x(k/N) - Q'_x(k/N)\}^2 \\ +\{Q_y(k/N) - Q'_y(k/N)\}^2 \end{matrix}} \tag{12}$$

Eq. (12) is applicable only when the original Bezier curve Q(t) is given. The expected results can be calculated by choosing proper pair of points of the given and approximated curves.

In order to get the exact locations of the control points, the intermediate control points are moved alternately to one of the four directions (x±∂x, y), (x, y±∂y). If the minimum RMS error is obtained at location (x, y), the control points are not moved. But the step size for the distance of movement ∂ is being reduced to half of its previous value. The movement of the control points, $(P_1)^{j+1} \leftarrow (P_1)^j$ and $(P_2)^{j+1} \leftarrow (P_2)^j$, j=1, 2, . . . is not a geometric one. It may occur at any of the four directions depending on the contribution of the assuming location of the control point in the minimum RMS error calculation. So, these iterations are continued until RMS error becomes less than some permissible value or the learning iteration is larger than some permissible value (here it is 100). These values are decided experimentally [34]. This moving effect of control points is shown in Fig. 11.

If the step size ∂ is set to 1 then it takes more time as it increases the iterations. Although it gives accurate result for some Bezier curves but it reduces efficiency of the overall computation. Modifying the step size to variable step size ∂x=∂y=5, 50, 75 and 100 has shown to gives almost 100% accuracy for most of the Beziers. Additionally, it increases efficiency by reducing number of iterations [34]. It has been observed that the height assumptions H & step sizes ∂ are working absolutely fine for curve with any orientation. So, the H & ∂ are kept unmodified in proposed algorithm. The modified algorithm followed in the second stage is given below:

[Step 1]

- Initialization: allsteps [] = {5, 50, 75,100}
- Count = 0 (Learning time)
- MAX = 100 (Maximum learning time)

```
• Stepdetermine r= 0 (Variable to determine step)
• ∂x = ∂y = allsteps [stepdeterminer] (Variable displacement for the movement of the control points)
• x = assumed_x; y = assumed_y (Initial assumption of control points at 1 st stage)
      ε = 0.000001 (Minimum permissible RMS error)

[Step 2]
• Count++; if (Count>MAX), then goto step 3
• if E (x + ∂x, y) or E (x - ∂x, y) is minimum then goto step 3
• if E (x, y + ∂y) or E (x, y - ∂y) is minimum then goto step 4
• if E (x, y) is minimum then ∂x = ∂x /2, ∂y = ∂y /2, goto step 2

[Step 3] Searching minimum RMS error for x-direction
• while (E (x + ∂x, y) < E (x, y) {
•         Count++, x = x + ∂x }
• while (E (x - ∂x, y) < E (x , y) {
•         Count++, x = x - ∂x }
• if (Count > MAX || (x, y) < ε), then goto Step 5 else goto Step 2

[Step 4] Searching minimum RMS error for y-direction
• while (E (x, y + ∂y) < E (x , y)  {
•         Count++, y = y + ∂y }
• while (E (x, y - ∂y) < E (x , y) {
•         Count++, y = y - ∂y }
• if (Count > MAX || E(x, y) < ε), then goto Step 5 else goto Step 2

[Step 5]
• if(E(x, y) > ε && stepdeterminer < allsteps.Length) {
• //trying with another step
• Count = 0; stepdeterminer++; ∂x = ∂y = allsteps[stepdeterminer];
• x = assumed_ x; y = assumed_y; goto Step 2
• }
• Else {  Exit   }.
```

The above algorithm gives us almost exact locations of control points for a Bezier. Fig. 12 shows one example of control points generated by for input Bezier curve. These control points are further considered for feature generation like FD.

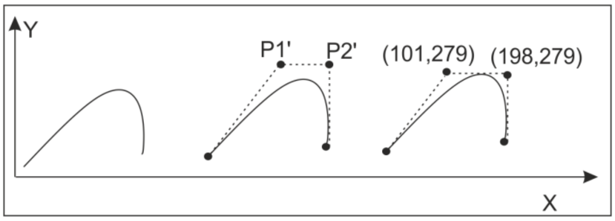


Figure. 12 Original input curve, assumed control points in 1st stage, and recovered control points in 2nd stage

5. Shape signatures and shape indexing using Fourier descriptions

In general, a shape signature is any 1-D function representing 2-D areas or boundaries. Four shape signatures are considered in this paper, these are central distance, complex coordinates (position function), curvature and cumulative angular function.

These four shape signatures for test and comparison are mostly used in recent FD implementations. They have been proved efficient for general shape representation. So, here also they are used. Now, assume the shape boundary coordinates have been extracted in the pre-processing stage.

A. Image Representation in the Logical Image Database

Image representation in Logical Image Database System is based on different image models. An image object is either specified in shape or texture pattern of the image. The logical image description includes: color, texture, and shape attributes. Texture attributes of grey level images are spatial deterministic or stochastic aspects of the grey level distribution [35]. Texture is an attribute of a region that is perceived as belonging together.

B. Logical Image Database:

A *shape signature* is any 1-D function representing 2-D areas or boundaries. In the following, assume the spline control point coordinates (x (t), y (t)), t = 0, 1… L-1, have been extracted in the pre-processing stage.

C. Euclidean distance

The Euclidean distance function is expressed by the distance of the control points from the first control point (x_p, y_p) of the spline curve.

$$r(t) = \sqrt{\left[x(t) - x_p\right]^2 + \left[y(t) - y_p\right]^2} \qquad (13)$$

Due to the subtraction of initial control point, which represents the starting position of the spline curve, from control points, the Euclidean distance representation is also invariant to translation and rotation. Here (x_p, y_p) is the first control point. Thus, by using above image categorization, the logical image database consists of only link of the image added in database [36].

D. Extraction of image features

An image feature is a piece of semantic information extracted from the image. There are several properties for measuring the quality of a feature:

Capacity: The number of distinguishable images that can be represented.

Maximal Match Number: The maximum number of images a query could possibly retrieve.

Complexity: The amount of computation required to determine if two images are similar for a specific feature.

Compactness: The amounts of space required to store and compare the feature.

Image features can be categorized as either *primitive* or *logical*. A primitive feature is a low-level or statistical attribute of an image such as an object boundary or color histogram. Primitive features are automatically extracted directly from the image. A logical feature represents an abstract attribute such as the label *grass* assigned to a region of an image. Logical features rely on information beyond that contained in the image. Feature extraction is performed when an image is added to the database. The system extracts feature that support image query types for shape & texture.

Fourier transformation on shape signatures is widely used for shape analysis; there are also some recent attempts to exploit it for shape retrieval [37]. The Fourier transformed coefficients form the FD of the shape. These descriptors represent the shape of the object in a frequency domain. The lower frequency descriptors contain information about the general features of the shape, and the higher frequency descriptors contain information about finer details of the shape [38]. Although the number of coefficients generated from the transform is usually large, a subset of the coefficients is enough to capture the overall features of the shape. The very high frequency information describes the small details of the shape, it is not so helpful in shape discrimination, and therefore, they can be ignored. As the result, the dimensions of the FD used for indexing shapes are significantly reduced [37].

Here the central distance function which is mentioned above in Eq. (13) is calculated for extracted control points and four extreme points as shown in Fig. 5 (a), representing dominant points of the curve. This central distance is our shape signature for computing the optimal feature vector that is FD. For a given shape signature described in this Section, $s(t)$, t = 0, 1, …, L-1, assuming it is normalized to N points in the sampling stage, the discrete Fourier transform of $s(t)$ is given by

$$u_n = \frac{1}{N}\sum_{t=0}^{N-1} s(t)e^{\left(\frac{-j2\pi nt}{N}\right)}, \mathrm{n} = 0, 1, 2, \ldots \mathrm{N}-1 \quad (14)$$

The coefficients u_n, n = 0, 1… N-1, are usually called Fourier descriptors (FD) of the shape, denoted as FD_n, n = 0, … N-1 [36].

6. Implementation and results

The pre-processing gives unsorted set of curve points using gray scale conversion, binarization, denoising, boundary detection Algorithms. This curve is boundary of a shape or object. Now to make the approach position invariant, center of boundary of every shape is aligned to origin. Then it is converted to linear curve form. This linear curve is normalized for achieving rotational invariance. The normalized curve is further divided into 10 nearly equivalent segments using segmentation algorithm. Each of these 10 segments can be considered as a separate Bezier curve. These curve points are now first arranged in sorted order of their tracing path. The modifications proposed in section 3 to control point generation make algorithm efficient. Pin pointed approach detects control points in a smaller number of iteration than old algorithm. The generated control 4 control points in addition to extreme points of the curve are considered in this approach to compute the FD as an optimal feature vector. For each segment 2 such a FDs are calculated. All these 20 FDs w. r. t. each segment is stored in database. So, a complete image can be represented by set of 20 FDs and few selected points. This reduces storage requirement by a large factor. Obviously, such representation leads to efficient comparison at time of matching query images.

After this the query image from user can be accepted. The above-mentioned procedure is repeated to find the FDs of the query image. Then the FDs of the query image are compared with database. If nearest match found then the image from the database is retrieved as a result. All steps of proposed method are shown Fig. 13.

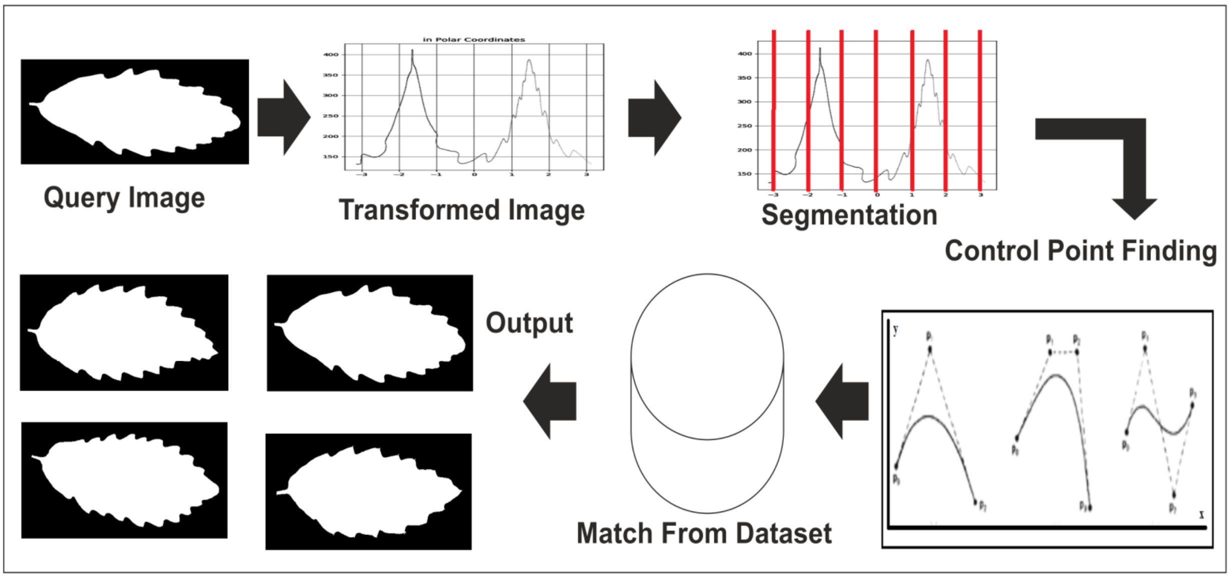


Figure. 13 Algorithm flow

Table 1. Segment 1 selected points for FD calculation

Image	Segment no	Selected control & Dominant Points (r)	Selected control & Dominant Points (θ)
	1	0.0989	149
		0.0806	147
		0.00606	143
		-0.0806	138
		-0.164	136
		-0.292	132
		-0.420	148
		-0.414	149
		-0.411	158
		-0.550	163

Table 2. FD points and selected points (in polar form) for input image with all 10 segments

Image	Segment no	FD1	FD2
	1	98.0121	214.898
	2	39.028051	856.666001
	3	62.595069	1375.003005
	4	58.42355	1281.845115
	5	34.698799	756.163285
	6	26.373986	583.818349
	7	37.119295	827.634739
	8	58.880703	1306.046822
	9	57.095277	1265.23953
	10	35.100282	777.964658

Table 1 shows selected points from segment 1 of the input image. These points are shown in polar coordinate system (r) form. Some values of ‘r’ are negative as the input image is aligned to origin. This implies that image is spread over all quadrants of coordinate system. Table 2 shows sample output of FDs calculated for an input image with all 10 segments.

Most commonly used UCI [39] Leaf Dataset is selected to test the average retrieval accuracy of the proposed method. It contains 340 images of 40 different species of plants. The images are 720 X 920 pixels, resolution and in 24-bit RGB colour model.

Evaluation Parameters

Performance of proposed method is evaluated by using four types of evaluation parameters 1) Accuracy, 2) Precision, 3), Sensitivity or Recall 4) Specificity. Precision and recall are the most widely used evaluation parameters for measuring performance of CBIR Systems. The evaluation parameters are mathematically represented using following terms:

- True positive (TP): It denotes the images retrieved correctly as belonging to the class.
- False Positive (FP): It denotes the images incorrectly retrieved as belonging to the class.
- True negative (TN): It denotes the images correctly retrieved as not belonging to the class.
- False negative (FP): It denotes the images incorrectly retrieved as not belonging to the class.

1) Accuracy: It is the ratio of the retrieved results that is true positive and false negatives to the total number of retrieved images. It can be mathematically expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

2) Precision: It is defined as the ratio of relevant output instances retrieved to the total instances retrieved. Mathematically, it can be defined using Eq. (16):

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

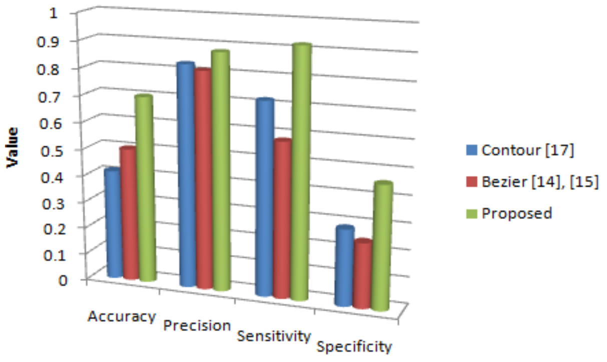
3) Sensitivity: It is the ratio of obtained relevant images retrieved to the total number of relevant images present in the database. It is also called as recall. It can be mathematically represented as:

$$Sensitivity = \frac{TP}{TP + FN} \quad (17)$$

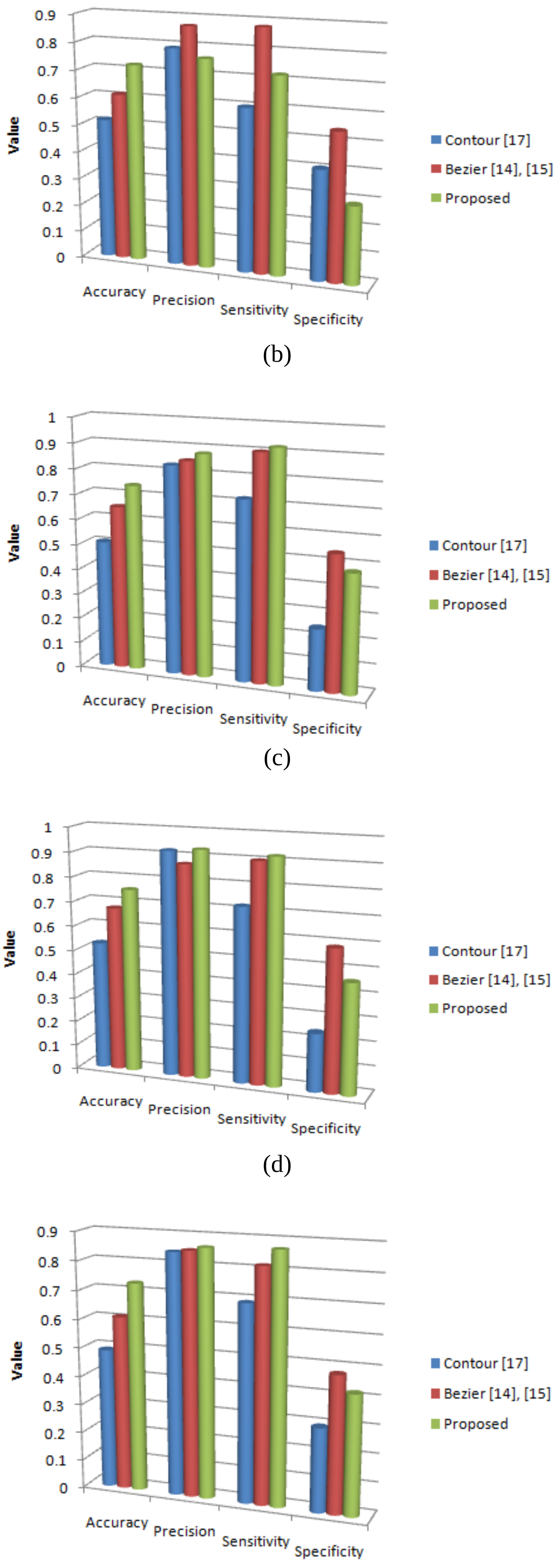
4) Specificity: It is the proportion of truly negative class predicted over the total false class and mathematically given as:

$$Specificity = \frac{TN}{TN + FP} \quad (18)$$

The performance of the proposed method is compared with those of various state-of-the-art shape-based image retrieval methods, classified as contour based [17], and Bezier curve-based image retrieval techniques [14, 15]. Multiple iterations are performed on dataset with different samples. Then results are averaged out as mean results. These results are compared with existing techniques and are shown in Fig. 14 in graphical form. Proposed method has high specificity with acceptable accuracy and sensitivity. Other methods Contour [17] and purely Bezier [14, 15] fall short in comparison to proposed method. In case of sensitivity and specificity Contour method out performs proposed method. But in case accuracy



(a)



(b)

(c)

(d)

(e)

Figure. 14 Results from leaf dataset: (a) iteration1, (b) iteration2, (c) iteration3, (d) iteration4, and (e) mean of all iterations

and specificity proposed method outperforms both methods. The proposed method outperforms in processing time due to the efficient method used for control point generation and also reduced processing time in feature matching due to the use of FDs.

7. Conclusion

This paper has designed and implemented a system that evaluates the similarity of each image in its database to a query image in terms of shape characteristics, and returns the images within a desired range of similarity.

Here, novel way of using control points as shape signature of an object is proposed. To detect the control points new and efficient approach is used. Results show it has acceptable accuracy. Generated control points are further transformed into optimal Feature vector Fourier Descriptor (FD).

In this approach an image is represented using only FDs and few selected points. Thus, size of features which are finally stored in database is substantially reduced. It further reduces the computation overhead at the time of feature matching for a query image. Additionally, this approach is position and orientation invariant. It works fine for similar objects with any orientation and irrespective of its position.

Due all these reasons proposed method outperforms others in terms of computation complexity and memory efficiency. Results prove that proposed method achieves feature selection accuracy.

For image databases of restricted variety, which merely includes homogeneous patterns, this approach yields satisfactory results. The performance and results obtained in our system are better as compared to other CBIR systems. The proposed approach finds its application in many areas such as; Trademark recognition, Medical diagnosis for tumor detection, Character recognition, Online digital signature matching, Remote sensing application, Handwriting recognition, Ecological learning of leaf, flower, petal etc.

References

[1] K. Kruthika, Rajeshwari, H. Maheshappa, and Alzheimer's Disease Neuroimaging Initiative, “CBIR system using Capsule Networks and 3D CNN for Alzheimer’s disease diagnosis”, *Informatics in Medicine Unlocked*, Vol.14, pp. 59-68, 2019.

[2] K. Chatfield, J. Philbin, and A. Zisserman, “Efficient Retrieval of Deformable Shape Classes using Local Self-Similarities”, In: *Proc. of 2009 IEEE 12th International Conf. on Computer Vision Workshops*, pp. 264-271, 2009.

[3] Y. Mei and D. Androutsos, “Robust Affine Invariant Region-Based Shape Descriptors: The ICA Zernike Moment Shape Descriptor and the Whitening Zernike Moment Shape Descriptor”, *IEEE Signal Processing Letters*, Vol.16, No.10, pp. 877-880, 2009.

[4] J. Shotton, A. Blake, and R. Cipolla, “Contour-Based Learning for Object Detection”, In: *Proc. of 10th IEEE International Conf. on Computer Vision*, Vol.1, pp. 503-510, 2005.

[5] X. Bai, X. Yang, and L. Latecki, “Detection and recognition of contour parts based on shape similarity”, *Pattern Recognition*, Vol.41, No.7, pp. 2189-2199, 2008.

[6] B. Zhong and W. Liao, “Direct Curvature Scale Space: Theory and Corner Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.29, No.3, pp. 508-512, 2007.

[7] Y. Cui and B. Zhong, “Shape retrieval based on parabolically fitted curvature scale-space maps”, *Intelligent Science and Intelligent Data Engineering*, Lecture Notes in Computer Science, Vol.7751, pp. 743-750, 2013.

[8] Y. Xia, L. Zhu, X. Luo, and H. Bae, “Automatic Matching Method of Control Points for AVHRR Remote Sensing Image”, *Advanced Science and Technology Letters*, Vol.45, pp. 18-23, 2014.

[9] S. Salve and K. Jondhale, “Shape Matching and Object Recognition Using Shape Contexts”, In: *Proc. of 2010 IEEE 3rd International Conf. on Computer Science and Information Technology*, Vol.9, pp. 471-474, 2010.

[10] R. Hemalatha, V. Vijaybaskar, and T. Thamizhvani, “Performance Evaluation of Contour Based Segmentation Methods for Ultrasound Images”, *Advances in Multimedia*, Vol.2018, pp. 4976372:1-4976372:8, 2018.

[11] R. Khan, T. Raisa, and R. Debnath, “An Efficient Contour Based Fine-Grained Algorithm for Multi Category Object Detection”, *Journal of Image and Graphics*, Vol.6, No.2, pp. 127-136, 2018.

[12] J. Kosinka, M. Sabin, and N. Dodgson, “Control vectors for splines”, *Computer-Aided Design*, Vol.58, pp. 173–178, 2015.

[13] J. Coussement, K. Steppe, P. Lootens, I. Roldán-Ruiz, and T. Swaef, “A flexible geometric model for leaf shape descriptions with high accuracy”, *Silva Fennica*, Vol.52, No.2, pp. 7740:1-7740:14, 2018.

[14] Y. Chi, C. Chien, and T. Lin, “Leaf Shape Modeling and Analysis Using Geometric Descriptors Derived from Bezier Curves”, *Transactions of the American Society of Agricultural Engineers*, Vol.46, No.1, pp. 175-185, 2003.

[15] S. Zube, “Interpolation method for quaternionic-Bézier curves”, *Lietuvos matematikos rinkinys, Proc. of the Lithuanian Mathematical Society, Ser.A*, Vol.59, pp. 13-18, 2018.

[16] H. Lang, J. Zhang, Y. Xi, X. Zhang, J. and Meng, “Fast SAR Sea Surface Distribution Modeling by Adaptive Composite Cubic Bézier Curve”, *IEEE Geoscience and Remote Sensing Letters*, Vol.13, No.4, pp. 505-509, 2016

[17] P. Chaisuk, K. Phromsuthirak, and V. Areekul, “Leaf Classification Based on A Quadratic Curved Axis”, In: *Proc. of 2017 IEEE International Conf. on Image Processing*, pp. 4472-4476, 2017.

[18] C. Willcocks, P. Jackson, C. Nelson, and B. Obara, “Extracting 3D Parametric Curves from 2D Images of Helical Objects”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.39, No.9, pp. 1757-1769, 2017.

[19] J. Russ, *The Image Processing Handbook*, 6th Edition, CRC Press, Boca Raton, FL, 2016.

[20] P. Wong and Z. Abas, “An Analysis of Human Silhouettes with Normalised Silhouettes Images and Shape Fourier Descriptors”, *International Journal of Human and Technology Interaction*, Vol.1, No.1, pp. 31-35, 2017.

[21] K. Gaurav and P. Bhatia. “A detailed review of feature extraction in image processing systems”, In: *Proc. of 2010 IEEE 4th International Conf. on Advanced Computing & Communication Technologies*, pp. 5-12, 2014.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, “Ssd: Single shot multibox detector”, In: *Proc. of European conf. on computer vision, Lecture Notes in Computer Science*, Vol.9905, pp. 21-37, 2016.

[23] S. Milan, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 4th Edition, Cengage Learning, Boston, MA, 2014.

[24] S. Gong and T. Newman, “Fine feature sensitive marching squares”, *IET Image Processing*, Vol.11, No.9, pp. 796-802, 2017.

[25] S. Gong and T. Newman, “Dual Marching Squares: Description and Analysis”, In: *Proc. of 2016 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 53-56, 2016.

[26] M. Misro, A. Ramli, J. Ali, and N. Hamid, “Cubic Trigonometric Bezier Spiral Curves”, In: *Proc. of 2017 14th International Conf. on Computer Graphics, Imaging and Visualization*, pp. 14-20, 2017.

[27] F. Gerald, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 5th Edition, Morgan Kaufmann Publishers Inc. San Francisco, CA, 2014.

[28] R. Pressman, *Software Engineering: A Practitioner's Approach*, 5th Edition, Tata-McGraw Hill Publications, New York, N.Y., 2001.

[29] N. Ansari and E. Delp, “On Detecting Dominant Points”, *Pattern Recognition*, Vol.24, No.5, pp. 441-451, 1991.

[30] R. Luo, Y. Perng, and P. Cheng, “Reverse engineering with trajectory generation based on Bezier curve at dual drive machine”, In: *Proc. of 2016 IEEE 14th International Workshop on Advanced Motion Control*, pp. 539-544, 2016.

[31] L. Howell and B. Allen, “Spline Trajectory Algorithm Development: Bézier Curve Control Point Generation for UAVs.”, In: *Proc. of 8th AIAA Atmospheric and Space Environments Conf.*, pp. AIAA 2016-4197:1-6, 2016.

[32] M. Khan, A. Chowdhury, S. Chowdhury, and K. Deb, “An Efficient Algorithm for Finding the Control Point of 3rd Order Bezier Curve”, In: *Proc. of IEEE 2010 International Forum on Strategic Technology*, pp. 441-445, 2010.

[33] M. Bhuiyan and H. Hama, “Recovering the control points of Bezier curves for line image indexing”, *Journal of Electron Imaging*, Vol.11, No.2, pp. 177-186, 2002.

[34] D. Watson, *Contouring: A Guide to the Analysis and Display of Spatial Data*, 1st Edition, Pergamon Press, London, 1992.

[35] S. Mahajan and R. Dogra, “A Review on Image Enhancement Techniques”, *International Journal of Engineering and Innovative Technology*, Vol.4, No.11, pp. 108-113, 2015.

[36] H. Moon, R. Chellappa, and A. Rosenfeld, “Optimal Edge-Based Shape Detection”, *IEEE Transactions on Image Processing*, Vol. 11, No. 11, November, pp. 1209-1227, 2002.

[37] X. Wang, S. Qiu, K. Liu, and X. Tang, “Web Image Re-Ranking Using Query-Specific Semantic Signatures”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.36, No.4, pp. 810-823, 2014.

[38] F. Ciompi, C. Jacobs, E. Scholten, M. Wille, P. Jong, M. Prokop and B. Ginneken, “Bag-of-Frequencies: A Descriptor of Pulmonary Nodules in Computed Tomography Images”,

IEEE Transactions on Medical Imaging, Vol.34, No.4, pp. 962-973, 2015.

[39] C. Mallah, J. Cope, and J. Orwell, “Plant Leaf Classification Using Probabilistic Integration of Shape, Texture and Margin Features”, In: *Proc. of 2013 International Conf. on Signal Processing, Pattern Recognition and Applications*, pp. 279-286, 2013.