A Multi-Level Polygonal Approximation Based Shape Encoding Framework for Automated Shape Retrieval

A project report submitted for the partial fulfillment of the

Bachelor of Technology Degree

in

Computer Science & Engineering

under

Maulana Abul Kalam Azad University of Technology

by

Sagnik Gupta

Roll No: 10402815040, Registration Number: 151040110751

Rahul Pramanik

Roll No: 10401615064, Registration Number: 151040110660

Academic Session: 2015-2019

Under the Supervision of

Prof. Sourav Saha

Department of Computer Science and Engineering
Institute of Engineering & Management

Y-12, Salt Lake, Sector 5, Kolkata, Pin 700091, West Bengal, India

Affiliated To

Maulana Abul Kalam Azad University of Technology

BF 142, BF Block, Sector 1, Kolkata, West Bengal 700064

May 2019

INSTITUTE OF ENGINEERING & MANAGEMENT
श्रद्धावान लभते ज्ञानम्
Salt Lake Electronics Complex, Kolkata - 700091, WB, INDIA

Phone : (033) 2357 2969/2059/2995
(033) 2357 8189/8908/5389
Fax : 91 33 2357 8302
E-mail : director@iemcal.com
Website : www.iemcal.com

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project report titled **"A Multi-Level Polygonal Approximation Based Shape Encoding Framework for Automated Shape Retrieval"**, submitted by **Sagnik Gupta, Roll No: 10402815040, Registration Number: 151040110751**, **Rahul Pramanik, Roll No: 10401615064, Registration Number: 151040110660**, students of **Institute of Engineering & Management** in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, is a bona fide work carried out under the supervision of **Prof. Sourav Saha** during the final year of the academic session of 2015-2019. The content of this report has not been submitted to any other university or institute for the award of any other degree.

It is further certified that the work is entirely original and the performance has been found to be satisfactory.

Prof. Sourav Saha
Assistant Professor
Department of Computer Science and Engineering
Institute of Engineering & Management

Prof.(Dr.) Himadri Nath Saha
H.O.D.
Department of Computer Science and Engineering
Institute of Engineering & Management

Prof.(Dr.) Amlan Kusum Nayak
Principal
Institute of Engineering & Management

Gurukul Campus: Y-12, Salt Lake Electronics Complex, Sector-V, Kolkata 700091, Phone: (033) 2357 2969
Management House: D-1, Salt Lake Electronics Complex, Sector-V, Kolkata 700091, Phone: (033) 2357 8908
Ashram Building: GN-34/2, Salt Lake Electronics Complex, Sector-V, Kolkata 700091, Phone: (033) 2357 2059/2995

INSTITUTE OF ENGINEERING & MANAGEMENT

DECLARATION
FOR NON-COMMITMENT OF PLAGIARISM

We, **Sagnik Gupta**, **Rahul Pramanik**, students of B.Tech. in the Department of Computer Science and Engineering, Institute of Engineering & Management have submitted the project report in partial fulfillment of the requirements to obtain the above noted degree. We declare that we have not committed plagiarism in any form or violated copyright while writing the report and have acknowledged the sources and/or the credit of other authors wherever applicable. If subsequently it is found that we have committed plagiarism or violated copyright, then the authority has full right to cancel/reject/revoke our degree.

Name of the Student: SAGNIK GUPTA

Full Signature: ____________________

Name of the Student: RAHUL PRAMANIK

Full Signature: ____________________

Date: ____________________

Abstract

With the ever-growing volume of image data, textual annotation of images for mining of query specific image has become impractical and inefficient. Thus, computer vision-based image retrieval has received considerable interest in recent years. One of the fundamental characteristics of an image-object is its shape which plays a vital role to recognize the object at a primitive level. Keeping this view as the central focus, we study the scope of a shape descriptive framework based on a multi-level polygonal approximation for generating shape defining features. Such a framework explores different degrees of convexity of an object's contour-segments and captures shape features at different approximation stages as the proposed algorithm determines polygonal approximations starting from coarse-level to more refined representation of a closed contour by varying number of polygon-sides. We have presented a shape-encoding scheme based on multi-level polygonal approximation which allows us to use the popular distance metrics to compute shape similarity score between two objects. The proposed framework, when deployed for similar shape retrieval task demonstrates fairly good performance in comparison with other popular shape-retrieval algorithms.

Acknowledgements

We must not forget to acknowledge everyone who has provided constant support to us during our B.Tech course. First and foremost, we would like to express sincere gratitude to our supervisor **Prof. Sourav Saha** for his continuous support and motivation in fueling the pursuance of carrying out this project endeavor. Without his guidance and persistent encouragement, this project work would not have been possible. He has been a tremendous mentor for us throughout this academic journey. Many of his academic advises about our career growth have been priceless.

We would like to convey sincere gratitude to **Prof. Himadri Nath Saha** for providing us constant inspiration to stand firm against several setbacks throughout the course. Additionally, we would like to thank all the technical, non-technical and office staffs of our department for extending facilitating cooperation wherever required. We also express gratitude to all of our friends in the department for providing the friendly environment to work on the project work.

We would also like to thank our Director **Prof. Satyajit Chakraborti** for providing us an outstanding platform in order to develop our academic career. In addition, we also preserve a very special thankful feeling about our Principal **Prof. Amlan Kusum Nayak** for being a constant source of inspiration.

A special thank is due to our family. Words cannot express how grateful we are to our parents for all the sacrifices that they have made while giving us necessary strength to stand on our own feet.

Finally, we would like to thank everybody who has provided assistance, in whatever little form, towards successful realization of this project but with an apology that we could not mention everybody's name individually.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Due to the recent developments in digital imaging technologies, an increasing number of images are generated every day which propels today's tech-savvy people to demand an automated system for retrieving images of their interest from large data-pool. Searching images using their textual annotations is a subjective process and is not practical for large databases. In recent years, shape is considered as one of the most promising criteria for automated identification of an object. In general, shape is an intuitive characteristic of an object which is widely understood yet difficult to define formally. The human perception of shape is a high-level intuitive concept whereas mathematical definitions tend to describe a shape with low-level features [1]. Many researchers tried to define a shape as a function of position and direction of simply connected curves within the two-dimensional field. Regardless of how shape gets defined, it has always been important visual information attracting the attention of researchers over the past few decades specifically dealing with the problems of computer vision. With the ever-growing demand of automated emulation of a human vision system, research interest among today's scientists in computer vision domain has been gradually driven towards obtaining maturity in automated shape analysis from both theoretical and practical point of view. A user survey in [2] indicated that 71% of the users were interested in object retrieval by shape. The overall motivation of this research work is to focus on exploring geometric properties of an object's closed contour for shape-based object retrieval.

1.2 Related Work

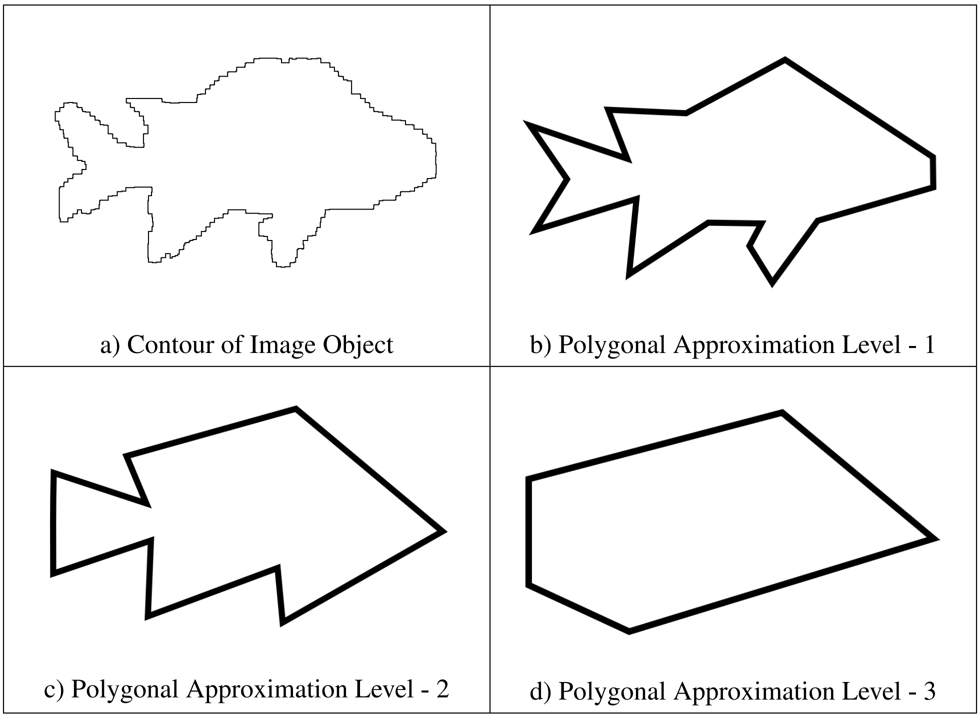
Shape-based image retrieval has received substantial attention from researchers over the past few decades. One of the earliest efforts made in solving this task is based on global features, such as invariant moments [3], Zernike moments [4, 5] geometric features [6], a combination of invariant moments and histogram of edge directions [7], and the angular radial transform descriptor [8]. In [9], a shape is partitioned into a set of closed contours and each contour is interpreted as a chain-code for facilitating shape matching. Another successful approach for shape retrieval relies on the construction of Attributed Relational Graph (ARG) [10, 11] by decomposing shape. An ARG of a shape is a fully connected graph, where a node denotes an interior region and edges manifest spatial relationships among the interior regions. For 2D shape matching, many approaches have been reported in the literature [2, 12]. Adamek and OConnor proposed an effective shape-descriptor that explores the degree of both concavity and convexity of the contour points [13]. They called it multi-scale convexity-concavity (MCC) representation, where different scales are rendered by smoothing the boundary using Gaussian kernels of varied widths. The curvature of a contour point is estimated by measuring the relative displacement with reference to its location in the preceding scale level. Afterward, the matching is performed by a dynamic programming (DP) approach. Belongie et al. [14] developed a shape descriptor termed as Shape Context. In Shape Context, a histogram is attached to each contour point for noting the relative distribution of the remaining points. The matching operation can be executed by finding the correspondence in a point-by-point manner. One of the most widely accepted shape representations is the curvature scale space (CSS) method proposed by Mokhtarian et al. [15, 16]. It has been recommended by the MPEG-7 community as the standard for boundary-based shape description [17]. In this method, the boundary is gradually smoothed using a set of Gaussian kernels resulting in approximated curves (CSS contours) until the boundary becomes totally convex. For matching, only the maximal inflection points of the CSS contours are used. The repeated application of Gaussian smoothing filters leads to the understanding of the contour evolution from coarse-level representation to fine-grained representation. Motivated by this idea, we have proposed a multi-level polygonal approximation of a contour by varying the number of sides of the shape-approximating polygon at each level.

1.3 Objective of the Proposed Work

The primary objective of the proposed work is to develop a multi-level polygonal approximation framework based on a sub-optimal thick-edged polygonal approximation method. The multi-level approximation strategy represents the contour with different degrees of approximation achieved by varying the number of sides of a shape-approximating polygon at each level (Fig. 1.1). To facilitate shape-based matching, a new shape encoding strategy along with an effective shape matching algorithm has been developed to quantitatively measure the shape similarity between two objects. The effectiveness of the proposed framework is shown by deploying it for the task of retrieving similar shapes with reference to a query object.

1.4 Organization of the Project Report

Rest of the report is organized as follows. In Chapter 2, the proposed technique is detailed focusing on a special thick-sided polygonal approximation, shape descriptor feature vector generation, and shape similarity score evaluation strategy. The subsequent chapter to the detailed framework-description has aimed to produce a report on how efficiently our framework has performed on shape-retrieval task by substantiating results. The concluding



a) Contour of Image Object

b) Polygonal Approximation Level - 1

c) Polygonal Approximation Level - 2

d) Polygonal Approximation Level - 3

Figure 1.1: Multi-level Polygonal Approximation

chapter summarizes the proposed work.

2

Proposed Framework

2.1 Overview of Proposed Framework

This section discusses our proposed framework in detail. The proposed model for shape retrieval initially extracts the contour of an image object using Moors Neighborhood boundary tracing algorithm. In the proposed framework, a shape is approximated at multiple stages by varying the number of sides of the approximating polygon. For a polygonal approximation of 2-D planar shape, we have applied sub-optimal thick-edged polygonal approximation method based on [18]. Fig. 2.1 intuitively demonstrates the overall flow of the proposed strategy. The description of each step of the overall flow is detailed next.

2.2 Polygonal Approximation

Extracting meaningful information and features from the contour of 2-D digital planar curves has been widely used for shape modeling [19, 20]. Detection of *dominant points (DP)* along the contour to represent visual characteristics of a shape has always been a challenging aspect for compact shape modeling. Polygonal approximation of a closed digital curve has always been considered as an important technique for compact shape

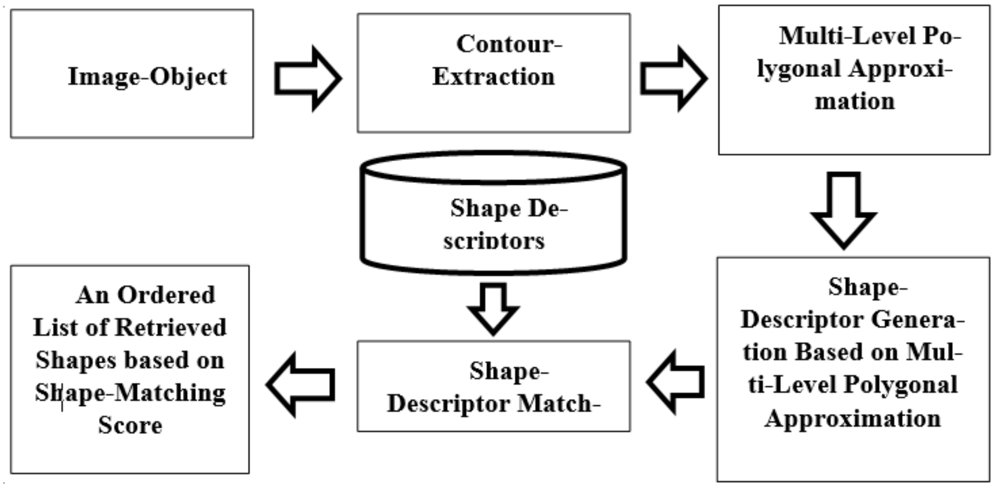


Figure 2.1: Overall Flow of the Proposed Scheme

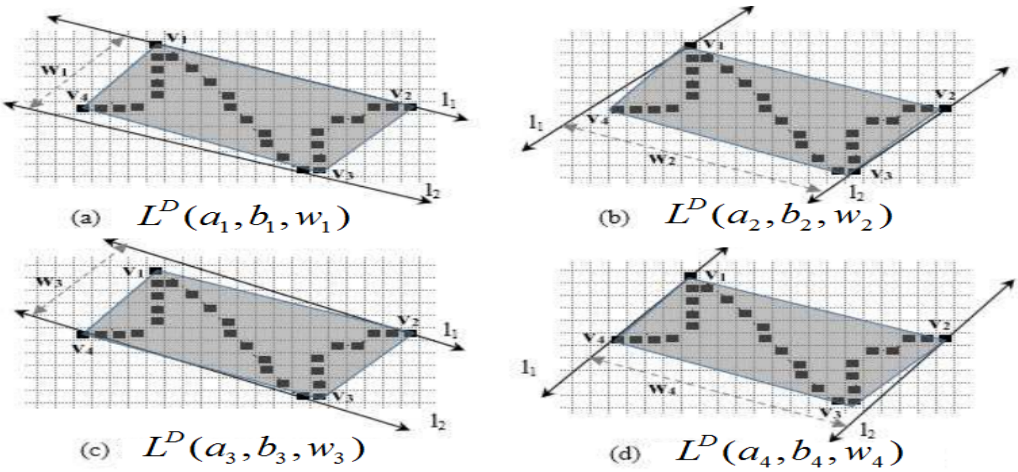


Figure 2.2: Thick Lines for a Discrete Point Sequence

modelling. We have formulated the problem of approximating a discrete curve in terms of optimally thick-line leading to design of an $O(n)$ algorithm for solving it.

2.2.1 Problem Formulation: Fitting Thick Line to a Discrete Curve

A discrete curve consists of a sequence of points in discrete space Z^2, where Z is the set of integers. In this chapter, a geometric model is used to describe a thick line for approximating a discrete curve by confining the sequence of curve points in discrete space Z^2. A few terminologies are defined below for illustrating the model conveniently with reference to Fig. 2.2.

Definition 1. A thick line, denoted by $L(a, b, w)$, is a pair of parallel lines defined by $l_1 : ax + by + c_1 = 0$ and $l_2 : ax + by + c_2 = 0$ where $w = \frac{|c_1 - c_2|}{\sqrt{a^2+b^2}}$ specifies its thickness.

Definition 2. A thick line $L^D(a, b, w)$ is for a sequence of discrete points $D(L)$ termed as a valid thick line if it confines $D(L)$ by satisfying the following condition.

$$D(L) = (x, y) \in Z^2 : (ax + by + c_1)(ax + by + c_2) \leq 0 \tag{2.1}$$

Definition 3. An optimal thick line $L^D_{opt}(a, b, w)$ for $D(L)$ is a valid thick line for which the thickness w is minimum. It can be mathematically expressed as below.

$$L^D_{opt}(a, b, w) = \arg\min_{w} L^D(a, b, w) \tag{2.2}$$

The pair of parallel lines (l_1, l_2) defining the thick line $L^D(a, b, w)$ for $D(L)$ essentially forms support lines of $D(L)$ and the distance between the pair corresponds to the thickness w. Using the above described model, the thick line fitting problem can be formulated as below.

Problem: Given a finite sequence of discrete points D, obtain $L^D_{opt}(a, b, w)$ by finding a pair of parallel lines such that

(1) *The pair forms support lines for D.*

(2) *The distance in between the pair of lines is the smallest possible distance to confine D.*

Observation 1. Given a sequence of discrete two dimensional points D and its convex-hull $CH(D)$, there exists a valid thick-line $L^D(a, b, w)$ defined by a pair of support lines (l_1, l_2) such that

(1) l_1 is coincident with an edge e_i of $CH(D)$.

(2) l_2 is another straight line passing in parallel with l_1 through a vertex v_k of $CH(D)$ which is farthest from e_i. The vertex v_k is termed as an antipodal vertex for edge e_i. The pair (v_k, e_i) can be termed as an antipodal vertex-edge pair and the distance $d(v_k, e_i)$ gives the measure of the antipodal distance for the pair (v_k, e_i).

Illustration 1. In Fig. 2.2, a valid thick line $L^D(a_i, b_i, w_i)$ of thickness w_i is defined by a pair of support lines $l_1 : a_i x + b_i y + c_{i1} = 0$ and $l_2 : a_i x + b_i y + c_{i2} = 0$ where $w_i = \frac{|c_{i1} - c_{i2}|}{\sqrt{a_i^2 + b_i^2}}$. It confines a sequence of points D such that each point of the sequence D lies in between the pair of support lines (l_1, l_2). A convex hull $CH(D)$ for the sequence D can also be generated with the vertices $\{v_1, v_2, v_3, v_4\} \in D$ in order to confine the sequence D. In the figure, one of the support lines of the valid thick-line $L^D(a_i, b_i, w_i)$ is coincident with one of the edges of the convex hull $CH(D)$. For example, in Fig. 2.2(a), l_1 is coincident with $e_1 = edge(v_1, v_2)$ and v_3 is antipodal vertex for e_1 as it is farthest from e_1 among all other vertices through which l_2 passes through. The antipodal distance $d(v_3, e_1) = w_1$ for the antipodal vertex-edge pair (v_3, e_1) can be considered as the thickness of the valid thick-line $L^D(a_1, b_1, w_1)$.

Observation 2. Given a sequence of discrete two dimensional points D and its convex-hull $CH(D)$, determining an optimal thick-line $L^D_{opt}(a, b, w)$ for D is equivalent of finding an antipodal vertex-edge pair $(v, e)_{opt}$ for which the antipodal distance is minimum. The observation can be expressed mathematically as below considering all possible antipodal vertex-edge pair of $CH(D)$.

$$\begin{aligned} (v, e)_{opt} &= \arg \min_{d(v_i, e_i)} \{(v_i, e_i)\} \\ \forall \quad & \textit{antipodal vertex-edge pair} \in CH(D) \end{aligned} \tag{2.3}$$

Illustration 2. In Fig. 2.2, as illustrated above a set of valid thick lines with different thickness confines a sequence of discrete 8-neighborhood-connected points ($D(L)$). Every thick line corresponds to an antipodal vertex-edge pair of the confining $CH(D)$. With close observation, it is found that the distance $d(v_3, e_1) = w_1$ for the antipodal vertex-edge pair (v_3, e_1) is minimum among all four antipodal vertex-edge pairs. This observation leads to the conclusion that the valid thick line $L^D(a_1, b_1, w_1)$ is an optimal thick line for $D(L)$ which corresponds to the antipodal vertex-edge pair (v_3, e_1).

2.2.2 Thick-Poly-line Approximation of Contour

In discrete geometry, a poly-line is a connected series of line segments. The poly-line is extensively used for approximating a digital curve with a polygon in order to represent its shape [21]. A poly-line is formally specified by a sequence of points called its vertices. There are many computer vision based applications based on shape analysis model wherein a digital curve representing various complex contour needs simplification. A digital curve can be effectively simplified by poly-line without loss of its inherent visual property. Techniques for the poly-line approximation of digital curve have been driving interest among the researchers for decades. The idea of poly-line approximation based on digital geometry has recently been explored extensively for simplified modeling of a shape [21, 22]. In this chapter, we have focused on developing an approximation strategy to determine the polygonal representation of a shape using special thick-poly-line wherein every line segment is an optimal thick line approximating a curve segment.

Cost of Fitting Optimal-Thick-line: Given a discrete curve segment with end points P_1 and P_N, its confining optimal thick line–$ThickLine(P_i, P_j)$ as illustrated before also associates a cost in terms of its thickness value for estimating qualitative aspect of the fitting process. The smaller thickness of the approximating line implies better fitting causing lesser cost. The cost associated with fitting a $Curve(P_i, P_j)$ using $ThickLine(P_i, P_j)$

is given below.

$$Cost(P_i, P_j) = Thickness(P_i, P_j) \tag{2.4}$$

Splitting Scheme: A curve is split into two segments at one of its convex-hull-vertices (Q_i) and the vertex at which the curve is split is termed as $PivotVertex$ in our scheme. The selection of $PivotVertex$ depends on optimal criteria as expressed through the following equations. The heuristic cost associated for choosing Q_i as a pivot-vertex is denoted as $SplitCost(P_1,\ P_N,\ Q_i)$ and it is computed based on thick-line-fitting-costs of the sub-segments generated on splitting the $Curve(P_1, P_N)$ at Q_i. The objective criteria for selecting $PivotVertex$ is to minimize aggregate costs of two split-segments as well as the difference between their individual costs. $SplitCost$ considers both aggregate-cost and cost-difference of two split-segments as expressed below. In Fig. 2.3, at root level $Q_1Q_2Q_3Q_4$ represents the vertices of the convex hull of a $Curve(P_1, P_N)$ and as per our scheme, the curve can be split at two convex-hull-vertices, namely Q_2 or Q_4, but Q_2 is selected as $PivotVertex$ because $SplitCost(P_1,\ P_N,\ Q_2)$ is less than $SplitCost(P_1,\ P_N,\ Q_4)$. The splitting scheme under repetitive application based on greedy Best-First-Heuristic exploration leads to the generation of a tree-like decomposition flow-structure as presented in Fig. 2.3.

$$\begin{aligned}
CostSum(P_1, P_N, Q_i) &= Cost(P_1, Q_i) + Cost(Q_i, P_N) \\
CostDiff(P_1, P_N, Q_i) &= ABS(Cost(P_1, Q_i) - Cost(Q_i, P_N)) \\
SplitCost(P_1, P_N, Q_i) &= CostSum(P_1, P_N, Q_i) + CostDiff(P_1, P_N, Q_i) \\
PivotVertex &= \arg\min_{Q_i} SplitCost(P_1, P_N, Q_i)
\end{aligned} \tag{2.5}$$

Greedy Best-First-Heuristic Based Exploration Strategy: Greedy Best-First-Heuristic process explores a search-tree by expanding the most promising node chosen according to the heuristic cost [23]. The above illustrated splitting scheme under greedy Best-

First-Heuristic based exploration generates a tree-like decomposition flow-structure as presented in Fig. 2.3. Each node of the decomposition tree represents a curve segment and undergoes a further splitting operation if the termination condition is not met. The termination of the repetitive splitting operation takes place whenever the number of leaves reaches the user-specified number of dominant points. Each leaf in the recursion tree denotes a yet-to-be decomposed curve segment. Under such exploratory repetitive splitting strategy, at every step until termination, we select a leaf-node representing a curve segment which is split on the next move. The selection of a node for subsequent exploration is performed based on a greedy best-first-heuristic [23] strategy which considers the most promising node with minimum $SplitCost(P_1, P_N, Q_i)$. The proposed best-first-heuristic strategy examines all leaf-nodes and selects a leaf-node whose heuristic cost is minimum irrespective of tree-levels. At every splitting step, two more children are created leading to the generation of two new curve segments for subsequent exploration.

Example 1. Fig. 2.3 represents a tree-like decomposition flow-structure rendered while tracing our proposed strategy to fit a given curve with five poly-lines and generates ten dominant points of five ploy-lines for approximating the curve. Our proposed strategy splits the original curve at node-1 into two curve segments represented at node-2 and node-3 respectively based on the previously described splitting-scheme. Since the cost of the curve at node-3 is found to be larger than that of node-1, the curve at node-3 is selected next for further decomposition. The sequential order in which the nodes in Fig. 2.3 would be selected for successive decomposition depends on a greedy best-first-heuristic strategy [23]. For a given curve as shown in Fig. 2.3, the exploration of our proposed algorithm would select candidate nodes in a specific sequential order driven by the greedy best-first-heuristic strategy. For the given example, node-1 is selected first and then node-3 is chosen as next decomposition candidate followed by the selection of node-5, and lastly, node-2 gets selected to undergo the splitting operation. Ultimately on termination, the proposed scheme produces five thick linear segments corresponding to

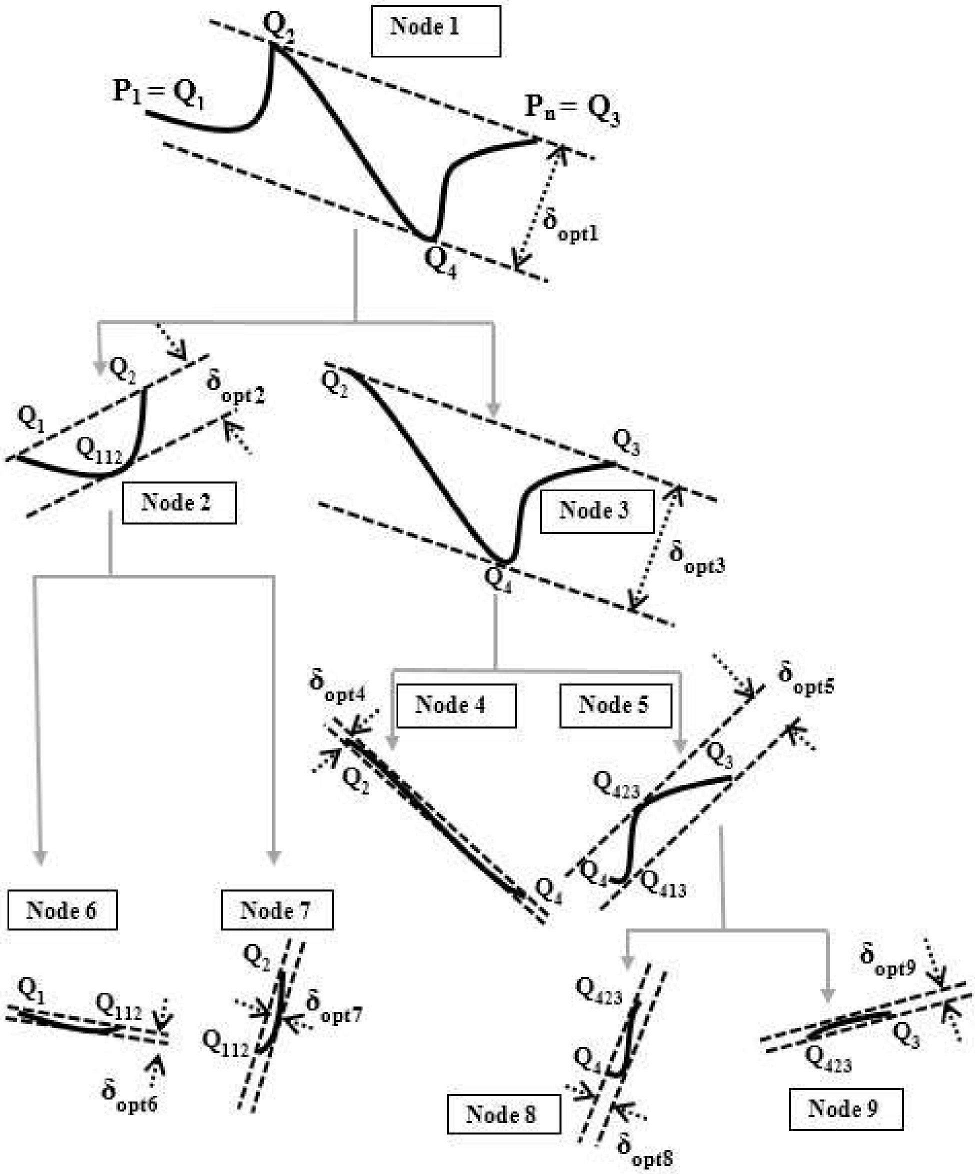


Figure 2.3: Greedy Best-First Splitting Strategy

five curve segments represented at node-4, node-6, node-7, node-8 and node-9 respectively as leaves of the decomposition tree. These five thick linear segments as a connected polyline approximate the given curve.

2.2.3 Proposed Algorithm

Algorithm 1: DOTHICKPOLYLINEAPPROX

Input: s and e: two end-points of a curve segment to be represented by poly-line and DP: number of dominant points to be obtained

Output: Global $PolyLineVertexList$: A set of points-pair each of which represents curve segment to be covered by poly-line

Add points-pair (s, e) to $PolyLineVertexList$;
$numSeg \leftarrow 1$;
while $numSeg < DP$ **do**
    $cost \leftarrow 0$;
    for *each points-pair* (P_i, P_j) *in* $PolyLineVertexList$ **do**
        if $cost < Cost(P_i, P_j)$ **then**
            $cost \leftarrow Cost(P_i, P_j)$;
            $s \leftarrow P_i$;
            $e \leftarrow P_j$;
        end
    end
    /* $Thickness(P_i, P_j)$ is determined based on Observation 2 */
    /* $ConvexHullVertexSet(Curve(s, e))$ is the set of convex-hull-vertices for $Curve(s, e)$ */
    $V \leftarrow ConvexHullVertexSet(Curve(s, e))$;
    $PivotVertex \leftarrow \arg\min_{Q_i \in V} SplitCost(s, e, Q_i)$;
    Remove points-pair (s, e) from $PolyLineVertexList$;
    Add points-pair $(s, PivotVertex)$ in $PolyLineVertexList$;
    Add points-pair $(PivotVertex + 1, e)$ in $PolyLineVertexList$;
    $numSeg \leftarrow numSeg + 1$;
end
return $PolyLineVertexList$;

The greedy best-first-heuristic algorithm (Algorithm 1: DOTHICKPOLYLINEAPPROX) is developed for approximating a closed digital curve with thick-poly-line and it is formally presented in this section. The digital curve is stored as an ordered list of two-dimensional points. Our proposed algorithm repetitively splits the curve leading to the generation of a tree like exploration as described earlier wherein each tree-node represents a curve segment. At every exploratory step, the best-first-strategy of the proposed method selects the most suitable node which is yet to be split and subsequently, the splitting scheme of the proposed model splits the curve segment corresponding to the selected node into two smaller segments. The repetitive splitting operation terminates whenever the number of leaves i.e. yet-to-be decomposed curve segments in the recursion tree reaches the user-specified desired number of dominant segments.

Average Time Complexity: The average time complexity of the algorithm for a curve of N points is given by the recurrence relation 2.6. At every exploratory step, the proposed thick-line fitting algorithm involves generation of a convex hull and determination of *optimal-valid-pair* for finding optimal thick-line. Convex-hull generation takes $O(N \log N)$ time complexity as we have used Graham-Scan [24] method while the determination of *optimal-valid-pair* requires $O(N)$ time complexity. Considering both, the proposed thick-line fitting strategy effectively causes $O(N \log N)$ time complexity. Additionally, at each step, the curve splitting heuristic scheme of the proposed model explores the vertices of convex-hull of the given curve for spitting which also involves the time complexity of $O(N)$. The average time complexity of the overall algorithm depends on recurrence relation 2.6 and approaches $O\left(N \log^2 N\right)$ for large N. The detailed derivation of recurrence relation 2.6 is presented in the appendix.

$$T(N) = \begin{cases} N \log N + \frac{1}{N} \times \sum\limits_{i=1}^{N} \{T(i) + T(N-i+1)\}, & \text{if } N > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

2.3 Multi-Level Polygonal Approximation

One of the interesting advantages of our poly-line approximation method is that we can apply it for a closed discrete curve which leads to the generation of a shape approximating polygon. In our proposed framework, users can flexibly specify the number of edges as an input parameter to obtain a thick-edged shape approximating polygon. The accuracy of the approximation increases if users specify more number of edges for approximating polygon. We assume that the degree of approximation varies inversely with the accuracy of shape approximation. This implies that the degree of approximation also varies inversely with the increase in the number of edges as evident in Fig. 2.5. We have utilized this

Approx/Thickness	10	20	50
No. of Sides	16	7	3

Figure 2.4: Multi-Level Polygonal Approximation

characteristic to approximate a shape at various approximation levels by varying number of edges of approximating polygon. It helps us to explore the evolutionary nature of the approximating polygons in capturing shape-detailing characteristics at different levels (Fig. 2.4). The average thickness of the edge depends on the specified number of edges of the polygon. As we increase the number of edges, the average thickness of the edges starts decreasing (Fig. 2.4). The average thickness of an edge of the shape approximating polygon corresponds to the degree of approximation.

2.4 Shape-Descriptor for Multi-Level Polygonal Approximation

As discussed earlier, each approximation level corresponds to a specific number of edges of the approximating polygon. Multi-level polygonal approximations are carried out repeatedly by specifying more number of polygon-edges at each iteration till the approximation error (i.e. average thickness of the edge) reaches a threshold value. At every stage of multi-level polygonal approximation, a shape descriptor feature vector is formed with respect to the corresponding shape approximating polygon. A special polygon-shape descriptive feature is formed based on its internal angles as illustrated below. While comparing two shapes at a specific level, the following sequence of steps

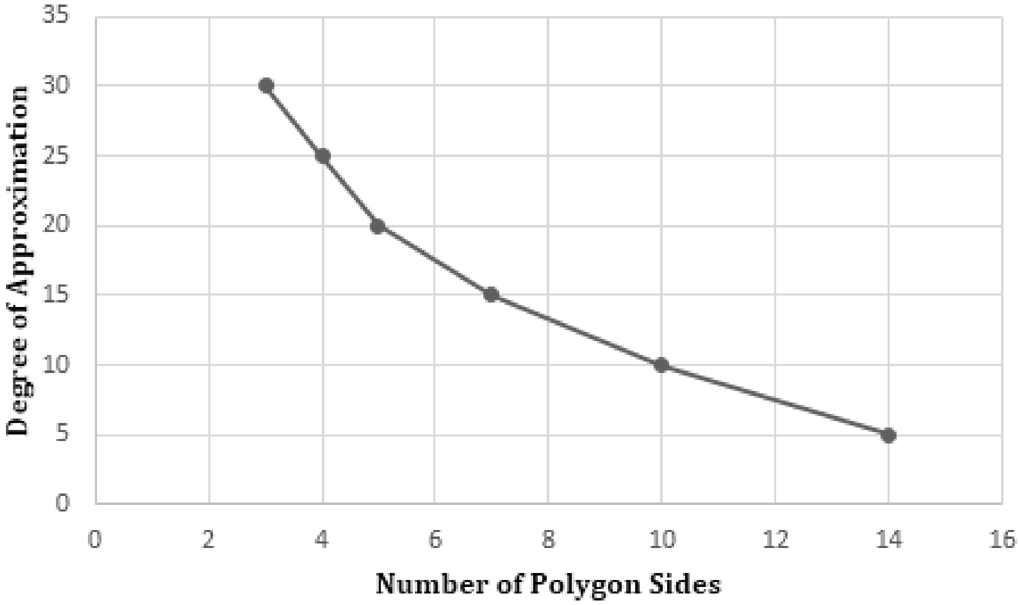
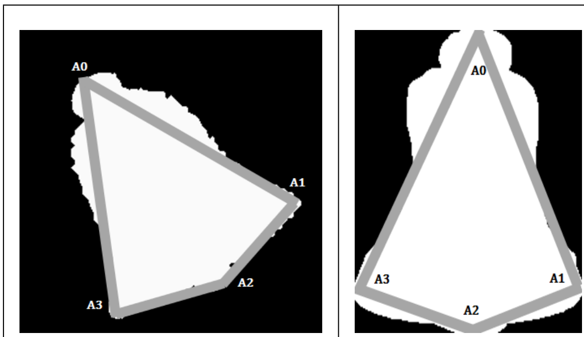


Figure 2.5: Degree of Approximation vs Number of Polygon-Sides



a) Shape-1 (N = 4) b) Shape-2 (N = 4)

Figure 2.6: Shape Approximating Polygon with sides = 4

is applied for forming shape-descriptive features and these features are used for shape-comparison. At each level, a shape dissimilarity score is computed based on Manhattan Distance between two shape-descriptive features. These steps are illustrated below with reference to Fig. 2.6a and Fig. 2.6b. Furthermore, Fig. 2.7 and Table 2.1 illustrate the shape descriptor generation strategy and matching scheme with reference to a few more shapes.

2.5 Basic Principle of the Shape-Descriptor Formation and Comparison Strategy

Step 1: The vertices of the shape-approximating polygon are indexed clockwise.

Step 2: Internal angle at each vertex (v_c) of the shape-approximating polygon is computed based on the angle needed to rotate edge(v_c, v_{c+1}) counterclockwise to align with edge(v_c, v_{c-1}) keeping vertex v_c fixed. Eqn. 2.7 is used to obtain internal angle $\theta_c°$ at vertex (v_c) with respect to adjoining vertices v_{c-1} and v_{c+1}.

$$\theta_{i,c} = \tan^{-1}(\frac{y_i - y_c}{x_i - x_c}) \times \frac{180°}{\pi}, i = c-1, c+1$$

$$f(\theta_{i,c}) = \begin{cases} 360° + \theta_{i,c} & \text{if } \theta_{i,c} < 360° \\ \theta_{i,c} & \textit{otherwise} \end{cases} \tag{2.7}$$

$$\theta_c = (360° + f(\theta_{c+1,c}) - f(\theta_{c-1,c})) \mod 360°$$

Step 3: Equation 2.8 is applied on internal angles of a shape-approximating polygon to compute a new set of angles. In case of Fig. 2.6a, the set will be {A0 = 31°, A1 = 99°, A2 = 126°, A3 = 102°}.

$$f(\theta_i) = \begin{cases} \theta_i° & \text{if } \theta_i < 180° \\ \theta_i° - 180° & \textit{otherwise} \end{cases} \tag{2.8}$$

Step 4: The new set of computed angles at the previous step are sorted in ascending order. In case of Fig. 2.6a, the ordered sequence will be {A0 = 31°, A1 = 99°, A3 = 102°, A2 = 126°}. In case of Fig. 2.6b, the ordered sequence will be {A0 = 42°, A1 = 88°, A3 = 97°, A2 = 131°}. These sequences are used as feature vectors at a specific level determined by the number of sides of the shape-approximating polygon.

Step 5: Manhattan Distance between two shape representative feature vectors is computed

Table 2.1: Shape Dissimilarity Score Matrix

Figure	Bell-1	Bell-2	Frog-1	Frog-2
Bell-1	0.0	13.5	39.5	36.8
Bell-2	13.5	0.0	34.7	34.6
Frog-1	39.2	34.7	0.0	9.8
Frog-2	36.8	34.6	9.8	0.0

and the distance is divided by the number of edges to obtain shape dissimilarity score at the respective level. The shape dissimilarity score for figures Fig. 2.6a and Fig. 2.6b is computed as 8.0 while the number of polygon-edges is 4.

Step 6: The final shape dissimilarity score between two shapes are determined by adding dissimilarity scores computed at different levels i.e. with varying edges (N) being 4, 8 etc. The final shape dissimilarity score for figures Fig. 2.6a and Fig. 2.6b is computed as 13.5 as shown in Fig. 2.7. Mathematically, the shape dissimilarity score between two shapes–S_1 and S_2 can be computed based on following Eqn. 2.9 where $f^i_{j,k}$ denotes j-th feature element of S_i at k-th approximation level.

$$Dissimilarity(S_1, S_2) = \sum_{k=4}^{N} (\frac{1}{k}) \sum_{j=1}^{k} |f^1_{j,k} - f^2_{j,k}| \tag{2.9}$$

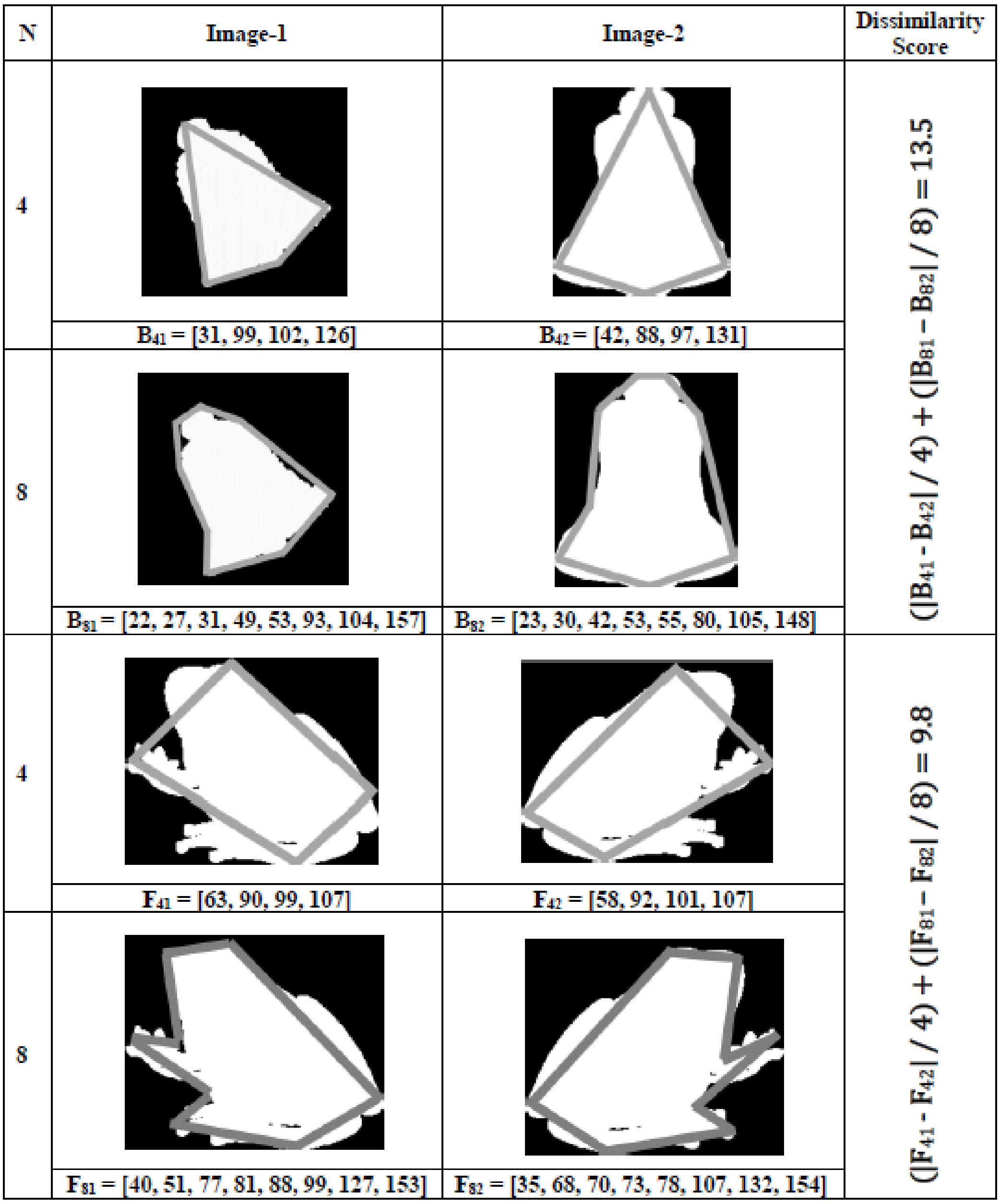


Figure 2.7: Feature Vector Formation at Multiple Stages

3

Experimental Results and Discussion

3.1 Experimental Setup

In order to evaluate the performance of the proposed shape retrieval framework, experiments have been conducted based on the MPEG-7 test database [17]. The dataset comprises 1400 shapes grouped into 70 classes and each class contains 20 similar objects. Some shapes of the same class have gone through various view transformations in addition to scaling, rotation, shearing process.

3.2 Experimental Results

Fig. 3.1 presents the retrieval result of the proposed framework based on a set of sample images from the MPEG-7 test database. It is evident from the observation that the proposed framework is able to retrieve most relevant images against a query image within the topmost three retrievals based on the illustrated similarity score ranking applied on respective feature codes of images. Table 2.1 presents the shape dissimilarity score matrix for two classes where each class contains two image objects. The results listed in Table 2.1 show that the dissimilarity score between any two objects belonging to the same class is smaller than the scores reported for the objects belonging to different classes. Fig. 3.2

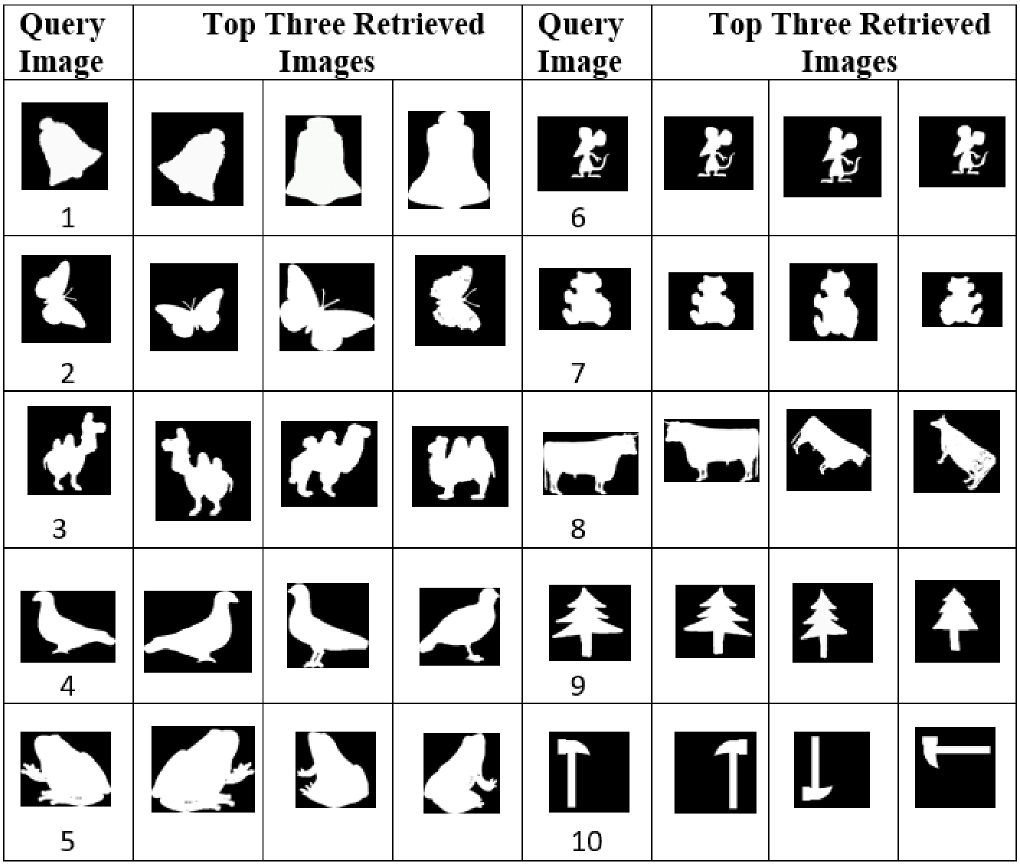


Figure 3.1: Retrieval Result

presents the execution-time profile of the proposed framework generated based on varying sides of a 32-sided star-polygon image object.

3.3 Performance Evaluation Metric

Evaluation of performance is difficult in content-based image retrieval because of the subjectivity of the human vision-based judgment. Ideally, the merit of a shape retrieval system depends on the relevance of the retrieved images ordered based on some similarity-score. However, many schemes for measuring the performance of a model have been attempted by researchers. Perhaps the most popularly used scheme for judging retrieval effectiveness is the Bull's eyes test [17]. Bull's eyes test provides us with a platform

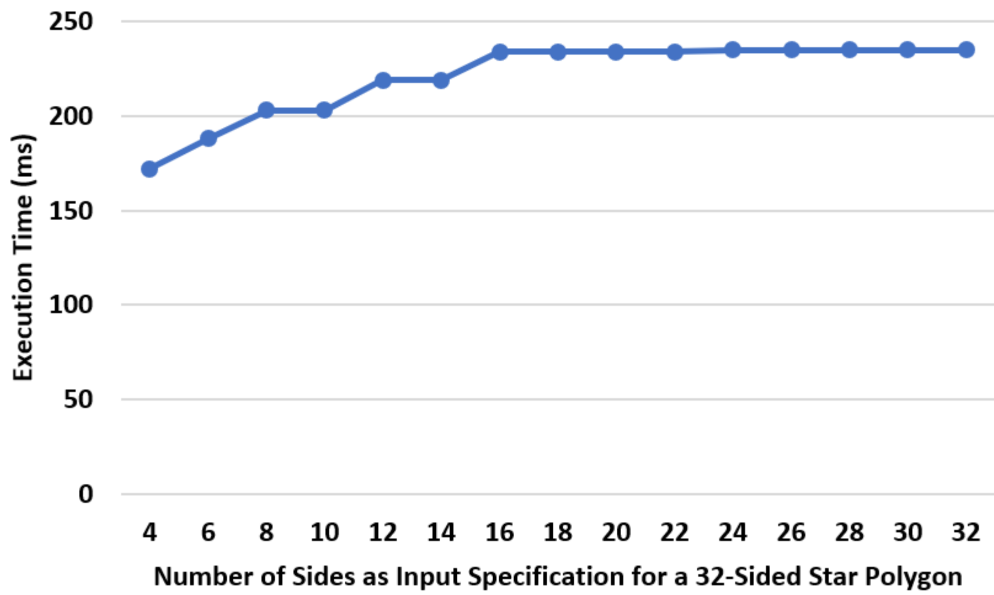


Figure 3.2: Execution Time Profile

for comparing the effectiveness of an approach in comparison with other shape retrieval approaches. Each shape of the dataset is compared with all other shapes, and among the top 40 retrieved shapes, we only consider the number of shapes retrieved by the applied model from the original class of the query shape. The Bull's eye retrieval rate for a query shape is computed as the ratio of the number of retrieved shapes from the class of the query shape to the total number of shapes contained by the query-shape's class. The overall Bull's Eye Percentage (BEP) score can be calculated by taking an average of all BEP scores obtained for different query shapes. Table 3.1 presents the performance of our proposed framework for the sample dataset listed in Fig. 3.1 in comparison with popular CSS scheme [15]. Every class of image dataset contains 20 samples and relevant retrievals are the shapes of the class to which the query shape belongs. The overall accuracy of the proposed framework is evaluated as 85% (Table 3.1) on Bull's eyes test, which seems good and comparable with the widely referred algorithm. In order to test the effectiveness of the shape descriptor vector with respect to the proposed similarity metric, we have also

Table 3.1: RESULT: Bull's Eye Score

Class Sample Id.	CSS Algorithm [15]		Proposed	
	Retrieval	BEP Score %	Retrieval	BEP Score %
1	17	85	17	85
2	18	90	17	85
3	18	90	17	85
4	16	80	16	80
5	19	95	18	90
6	15	75	18	90
7	18	90	18	90
8	16	80	15	75
9	18	90	15	75
10	19	95	19	95

performed K-NN classification on the data set and obtained fairly good class-prediction accuracy rate as evident from Table 3.2.

Table 3.2: KNN Classifier Result: Confusion Matrix (Prediction %)

Class Id	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	**97**	0	0	0	0	0	1	0	0	2
2	0	**96**	0	0	0	0	0	4	0	0
3	0	0	**97**	0	0	0	0	1	0	2
4	0	0	0	**97**	0	1	0	0	0	2
5	0	0	0	0	**97**	0	0	1	0	2
6	0	0	0	1	0	**96**	3	0	0	0
7	0	0	0	0	0	3	**97**	0	0	0
8	0	2	0	0	2	0	0	**95**	0	1
9	1	0	0	2	0	0	0	0	**96**	1
10	1	0	1	1	1	0	0	0	0	**96**

4

Conclusion

An effective shape descriptive framework called multi-level polygonal approximation is proposed for similar shape retrieval which explores different degrees of approximation of an object's contour while varying number of edges of the shape approximating polygon and a special shape encoding scheme has also been developed to measure shape similarity between two objects. The performance of the proposed framework is experimentally found to be good and comparable with existing state-of-the-art algorithms. However, we need to further analyze the robustness of the proposed framework under many other plausible deformations of closed contours.

Average Time Complexity Computation

$$T(N) = \begin{cases} N\log N + \frac{1}{N} \times \sum_{i=1}^{N} \{T(i) + T(N-i+1)\}, & \text{if } N > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

$$NT(N) = N^2 \log N + 2\sum_{i=1}^{N-1} T(i) \tag{4.2}$$

$$(N-1)T(N-1) = (N-1)^2 \log(N-1) + 2\sum_{i=1}^{N-2} T(i) \tag{4.3}$$

Subtracting Eqn. 4.3 from the Eqn. 4.2, we obtain

$$\begin{aligned} NT(N) - (N-1)T(N-1) &= N^2 \log N - (N-1)^2 \log(N-1) + 2T(N-1) \\ \text{i.e. } NT(N) &= N^2 \log N - (N-1)^2 \log(N-1) + (N+1)T(N-1) \\ \text{i.e. } NT(N) &= N^2 \log \frac{N}{N-1} + (2N-1)\log(N-1) + (N+1)T(N-1) \end{aligned}$$

We may assume without significant error that $\log \frac{N}{N-1}$ approaches zero for large N

$$\begin{aligned} \text{i.e. } NT(N) &\leq 2N\log(N-1) + (N+1)T(N-1) \\ \text{i.e. } \frac{T(N)}{N+1} &\leq \frac{2\log N}{N+1} + \frac{T(N-1)}{N} \end{aligned}$$

We obtain following expression by repeatedly substituting T(N - 1), T(N-2), T(N - 3),. . .

$$\text{i.e. } \frac{T(N)}{N+1} \leq \frac{2\log N}{N+1} + \frac{2\log(N-1)}{N} + \frac{T(N-2)}{N-1}$$

$$\text{i.e. } \frac{T(N)}{N+1} \leq \frac{2\log N}{N+1} + \frac{2\log(N-1)}{N} + \frac{2\log(N-2)}{N-1} + \frac{T(N-3)}{N-2}$$

....

....

$$\text{i.e. } \frac{T(N)}{N+1} \leq 2\sum_{k=1}^{N} \frac{\log k}{k+1} + \frac{T(0)}{1}$$

$$\text{i.e. } \frac{T(N)}{N+1} \leq 2\int_{1}^{N} \frac{1}{x}\log(x)dx$$

$$\text{i.e. } T(N) \leq 2(N+1)\log^2 N$$

$$\text{i.e. } T(N) = O\left(N\log^2 N\right)$$

Application Screen-Shots

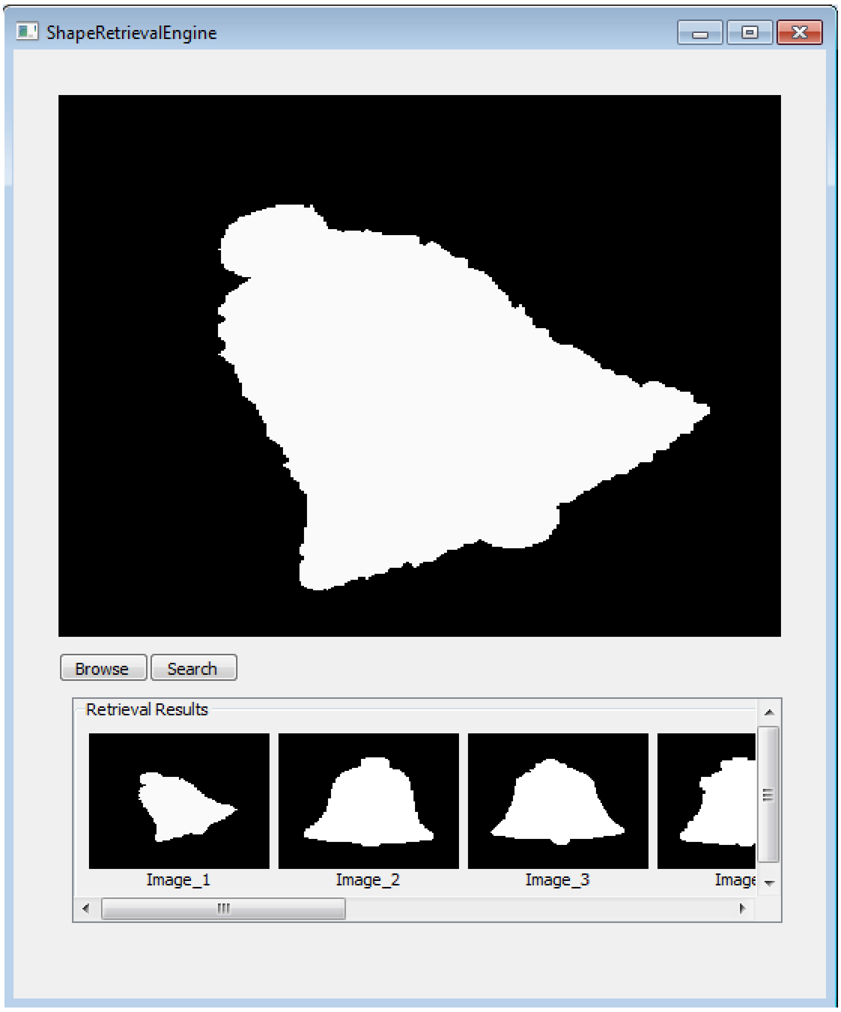


Figure 4.1: Application Screen-Shot

Bibliography

[1] S. Marshall, “Review of shape coding techniques,” *Image and Vision Computing*, vol. 7, no. 4, pp. 281–294, 1989.

[2] D. Zhang and G. Lu, “Review of shape representation and description techniques,” *Pattern recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[3] M. K. Hu, “Visual pattern recognition by moment invariants,” *IRE Trans. Information Theory*, vol. 8, pp. 179–197, 1962.

[4] A. Khotanzad and Y. H. Hong, “Invariant image recognition by zernike moments,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[5] Y. S. Kim and W. Y. Kim, “Content-based trademark retrieval system using a visually salient feature,” *Image and Vision Computing*, vol. 16, no. 12, pp. 931–939, 1998.

[6] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, “Efficient and effective querying by image content,” *J. Intelligent Information Systems*, vol. 3, no. 3, pp. 231–262, 1994.

[7] A. K. Jain and A. Vailaya, “Shape-based retrieval: A case study with trademark image databases,” *Pattern Recognition*, vol. 31, no. 9, pp. 1369–1390, 1998.

[8] W. Y. Kim and Y. S. Kim, “A new region-based shape descriptor,” *Technical Report ISO/IEC MPEG99/M5472*, 1999.

[9] H. L. Peng and S. Y. Chen, “Trademark shape recognition using closed contours,” *Pattern Recognition Letters*, vol. 18, no. 8, pp. 791–803, 1997.

[10] E. G. M. Petrakis, “Design and evaluation of spatial similarity approaches for image retrieval,” *Image and Vision Computing*, vol. 20, no. 1, pp. 59–76, 2002.

[11] E. G. M. Petrakis, C. Faloutsos, and K. Lin, “Imagemap: An image indexing method based on spatial similarity,” *IEEE Trans. Knowledge and Data Eng.*, vol. 14, no. 5, pp. 979–987, 2002.

[12] S. Loncaric, “A survey of shape analysis techniques,” *Pattern Recognition*, vol. 31, no. 8, pp. 983–1001, 1998.

[13] T. Adamek and N. E. OConnor, “A multiscale representation method for nonrigid shapes with a single closed contour,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 742–753, 2004.

[14] S. Belongie, J. Malik, , and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[15] F. Mokhtarian and M. Bober, *Curvature scale space representation: theory, applications, and MPEG-7 standardization*, vol. 25. Springer Science & Business Media, 2013.

[16] F. Mokhtarian and A. Mackworth, “Scale-based description and recognition of planar curves and two-dimensional shapes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 34–43, 1986.

[17] R. Ralph, “Mpeg-7 core experiment.” `http://www.dabi.temple.edu/~shape/MPEG7/dataset.html`, 1999.

[18] S. Saha, S. Goswami, and P. R. Sinha Mahapatra, “A heuristic strategy for sub-optimal thick-edged polygonal approximation of 2-d planar shape,” *International Journal of Image, Graphics and Signal Processing*, vol. 10, pp. 48–58, 2018.

[19] D. Zhang and G. Lu, “Review of shape representation and description techniques,” *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[20] K. Asrani and R. Jain, “Contour based retrieval for plant species,” *I.J. Image, Graphics and Signal Processing*, vol. 5, no. 9, pp. 29 – 35, 2013.

[21] R. Klette and A. Rosenfeld, *Digital geometry: geometric methods for digital picture analysis*. Elsevier, 2004.

[22] J. Mukhopadhyay, P. P. Das, S. Chattopadhyay, P. Bhowmick, and B. N. Chatterji, *Digital geometry in image processing*. CRC Press, 2013.

[23] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 ed., 2003.

[24] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.