An Efficient Descriptor-Based Approach for Dominant Point Detection in Shape Contours

Mohammad T. Parvez[*]

Department of Computer Engineering, Qassim University, 51477 Qassim, Saudi Arabia

[*] Correspondence: Mohammad T. Parvez (m.parvez@qu.edu.sa)

Abstract: Dominant points, or control points, represent areas of high curvature on shape contours and are extensively utilized in the representation of shape outlines. Herein, we introduce a novel, descriptor-based approach for the efficient detection of these pivotal points. Each point on a shape contour is evaluated and mapped to an invariant descriptor set, accomplished through the use of point-neighborhood. These descriptors are then harnessed to discern whether a point qualifies as a dominant one. Our proposed methodology eliminates the need for costly computations typically associated with evaluating candidate dominant points. Furthermore, our algorithm significantly outperforms its predecessors in terms of speed, relying solely on integer operations and obviating the necessity for an optimization phase. Experimental outcomes, derived from the widely used MPEG7_CE-Shape-1_Part_B, denote a minimum enhancement of 2.3 times in terms of running time. This implies that the proposed methodology is particularly suitable for real-time applications or scenarios managing shapes comprising a substantial number of points.

Keywords: Contour representation; Shape descriptor; Outline modeling; Dominant point detection; Point suppression

1 Introduction

Dominant points, also referred to as control points, are instrumental in delineating the contours of digital planar shapes [1, 2]. When connected via straight lines, these points generate a sequence of line segments that approximate an image's outline. This form of approximation, known as polygonal approximation (illustrated in Figure 1), is employed in various applications, including shape modeling [3] and recognition [4]. The primary rationale behind utilizing this approximation-based modeling approach is its proficiency in effectively managing contour noise [3, 5]. Furthermore, polygonal approximation offers a more concise representation of the contour [1]. As a result, there has been a considerable amount of research over several decades dedicated to developing efficient algorithms for polygonal approximation [3, 5–19].
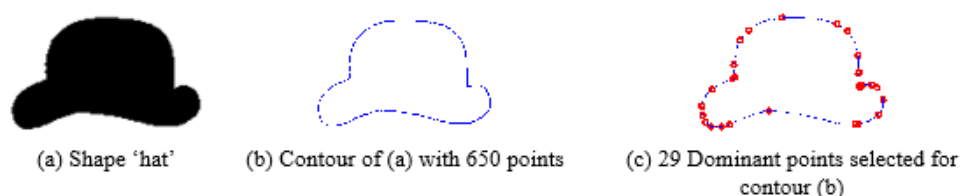

(a) Shape 'hat'   (b) Contour of (a) with 650 points   (c) 29 Dominant points selected for contour (b)

Figure 1. Illustration of dominant points extraction for a planar shape outline

Numerous polygonal approximation algorithms have been developed with the primary goal of identifying dominant points situated along the contour of a shape. These dominant points subsequently function as vertices in the construction of the approximating polygon [5, 10]. Typically, these approximation methodologies aim to minimize specific error criteria, such as the weighted mean squared error [8, 14, 20].

A prevalent approach shared by these algorithms involves establishing a significance measure for each point along the contour [18]. This measure is instrumental in guiding the selection of a subset of contour points,

either by emphasizing the most significant ones [7, 17] or by systematically eliminating less significant, redundant points [3, 5, 10].

The selection process necessitates a definition of the relative importance of contour points. This importance level can be determined using a predefined threshold [10] or by optimizing error measurements. It is noteworthy that research indicates optimization methods generally surpass threshold-based approaches in performance [3, 5]. The primary goal of these algorithms is to yield an accurate and efficient polygonal approximation that effectively encapsulates the fundamental characteristics of the shape contour.

This study aims to advance the current state-of-the-art in dominant point detection. Rather than relying on thresholds or costly optimization methods to select dominant points, it proposes employing intuitive, human-like heuristics. Humans often examine a point and its neighborhood to determine whether it is a dominant point. We aim to replicate this process algorithmically. Our method involves examining a point on a shape contour, using its neighborhood to ascertain whether it is part of a curve or a straight line. If the point is identified as part of a curve, it is retained for further investigation. If not, it is classified as a non-dominant point in a process we term 'point suppression'.

To this end, we have developed novel rotation-invariant shape descriptors that effectively encapsulate the shape outline. These descriptors can be computed without recourse to expensive computational operations. As a result, descriptor-based point suppression paves the way for a highly efficient dominant point detection algorithm, albeit at the cost of slightly less optimized approximation. To the best of our knowledge, this is the first study to use descriptors for dominant point detection in offline shapes.

The novelty of the proposed approach can be summarized as follows:

i. Innovative point suppression method using shape descriptors.

ii. Rapid algorithm that avoids costly optimizations compared to other methods.

iii. Utilization of adaptive local and global features for dominant point detection.

The paper is structured as follows: Section 2 gives an overview of the existing research on dominant point detection. Section 3 details the proposed algorithm. Section 4 provides a comprehensive analysis of the experimental results, and finally, Section 5 presents our conclusions.

2 Related Works

Numerous algorithms have been proposed for the detection of dominant points in digital planar curves, falling broadly into two main categories. The first category includes algorithms that strive to minimize the number of edges or vertices required to approximate the curve, subject to defined error criteria [21–23]. The second category, by contrast, comprises algorithms that pinpoint specific points along the curve as dominant points [3, 5, 7, 10–12, 14, 16, 17, 24].

Algorithms in the first category aim to achieve optimal digital curve approximations. However, these methodologies are often parametric, necessitating the determination of parameters such as the initial vertex or the maximum allowable error prior to generating an approximation. For instance, the method presented in the study [21] approximates a contour using a polygon with the minimum number of vertices, ensuring a predefined acceptable approximation error and an initial vertex. This algorithm iteratively identifies the smallest set of vertices for the polygonal approximation, commencing with a preselected initial vertex [18].

In contrast, algorithms in the second category typically exhibit increased robustness when applied to a variety of shapes. Given the plethora of dominant point detection algorithms chronicled in existing literature, our focus will be directed towards the more recent and relevant approaches [18]. Comprehensive evaluations of various dominant point detection algorithms can be found in the study [25].

A substantial number of dominant point detection algorithms initiate the process by applying filters to contour points based on various criteria. These may include orientation-specific predefined masks [7] or the detection of break-points [3, 10, 11]. Following this initial screening, each point on the contour, C, is allocated a significance measure, and the points deemed less significant are subsequently discarded.

Researchers have leveraged a myriad of techniques to compute this measure. For example, Chau and Siu [7] utilized cosine angles to define a significance measure for each potential dominant point. This measure was then employed to suppress less critical dominant points. In contrast, Masood [11] introduced the Associated Error Value (AEV) for each dominant point (DP), and an iterative process was applied to eliminate DPs with the lowest AEV. This was followed by an optimization procedure aimed at enhancing the Integrated Squared Error (ISE) of the resulting polygonal approximation. However, this scheme lacks a well-defined stopping criterion, which may limit its effectiveness.

Some researchers introduced the concept of a support region for each point along contour C [3, 10, 13, 24]. This support region for a point includes the points along C within a specific range defined by $\{p_{i-k}, \ldots, p_i, \ldots, p_{i+l}\}$ on C for some k and l. Subsequently, the significance measure for p_i is computed using mathematical operations that consider the support region of p_i. Consequently, less significant points can be removed from the set of dominant points, using the process call point-suppression [3, 10, 24].

The concept of point suppression was originally introduced by Marji and Siy [10]. Their approach, however, is constrained by a fixed suppression threshold. Carmona-Poyato et al. [24] refined this method by incorporating an optimization procedure to determine the threshold value for collinear-points suppression. Building upon their work, Carmona-Poyato et al. [5] further improved their technique by iteratively removing dominant points until a specific condition was met. Nonetheless, the performance of their algorithm can fluctuate significantly depending on the chosen final condition [18].

Both algorithms presented in studies [5, 24] fall into the realm of global optimization techniques and employ a single suppression parameter applicable to the entire contour. Conversely, the method presented in the study [3] highlighted the advantages of a local optimization strategy. This approach involves using distinct, adaptively estimated suppression thresholds for different segments of the contour, effectively taking into account the varying levels of detail within the contour [18].

Most of the algorithms discussed above determine the significance of a point based on its distance to the approximate straight line. This approach carries two primary disadvantages: firstly, the process of computing the significance in this manner is computationally intensive; secondly, a threshold is required to establish the cut-off distance. This study aims to circumvent these drawbacks by adopting a completely different approach: The significance measure of a point is estimated based on the shape formed around the point under consideration. As demonstrated in the experimental results section, this approach leads to a highly efficient dominant point detection algorithm.

3 Algorithm

The methodology proposed in this study is rooted in the use of invariant shape descriptors to characterize segments of a shape contour. We first elucidate how a shape can be represented using these descriptors. Following this, we discuss how these descriptors can be employed to suppress points along the shape outline.

3.1 Segment Descriptors

A shape contour (also known as an outline or stroke), C, is defined as an ordered sequence of n points (x_i, y_i), $i = 1, 2, \ldots, n$ [19]. Here, point (x_i, y_i) has two neighbors: (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}), $i = 2, 3, \ldots, n - 1$. In the case where shape C represents an online stroke, it's important to note that the endpoints in C will have only one neighboring point. In this section, we focus only on a sub-set S of C, such that $|S| = (m + 1) < n$. That is, S contains a set of consecutive points from C, where p_j is the middle point in S. The value $m = |S| - 1$ is considered as the region of support (as describe earlier) or strength for p_j. We now describe how this segment S of contour C is described by a shape descriptor.

Consider a set of descriptors, denoted as D, which characterizes the directional movement of points in S. Our objective is to construct D in such a way that every descriptor, represented as $d \in D$, remains invariant to rotation, translation, and scale changes. The specific descriptors employed in this study are outlined in Table 1. Consequently, each segment S can be described using one of three descriptors: straight_line, ccw_curve, and cw_curve.

Table 1. Rotation-invariant shape descriptors used in this work

Straight Line (straight_line)	—
Counter-clock-wise Curve (ccw_curve)	↖
Clock-wise Curve (cw_curve)	↗

The procedure for associating a segment S with a descriptor is explained next. To illustrate this process, please refer to Figure 2 and Figure 3. Subgraphs (a) and (b) as well as (c) and (d) of Figure 2 display a shape (referred to as 'camel' from the popular MPEG7_CE-Shape-1_Part_B database [26]) and the corresponding contour that requires modeling by a shape descriptor. Figure 3 demonstrates how a contour segment is mapped to a descriptor from Table 1.

As shown in the bottom-left corner of Figure 3, we partition the 2D planar space into eight regions. This division serves a purpose akin to the well-known quantization technique, allowing us to quantize a shape trajectory into a limited set of discrete descriptors [27]. Now, consider a segment S (extracted from subgraph (d) of Figure 2). Suppose we aim to determine the descriptor for the segment with endpoints $\{ep_i, ep_{i+1}\}$, as depicted in Figure 3.

To determine the appropriate descriptor from D that matches segment S, we introduce another point along S, which we'll call $mid_{i,i+1}$. This point represents the midpoint of the segment S. Now, given a tuple $P = <ep_i, mid_{i,i+1}, ep_{i+1}>$, corresponding to the segment S, we derive the descriptor $d \in D$ as follows:

We start by translating the points in P so that $mid_{i,i+1}$ becomes the origin. Then, we estimate the positions of ep_i and ep_{i+1} within the 2D space, following the subdivision shown in Figure 3. Let's denote these positions as

pos (ep_i) and pos (ep_{i+1}) respectively. Additionally, we determine the parameter $loc(mid_{i,i+1})$, which specifies the location of point $mid_{i,i+1}$ in relation to the directional line-segment stretching from ep_i to ep_{i+1}. It's worth noting that $loc(mid_{i,i+1})$ can take one of two values, 'left' or 'right', based on whether $mid_{i,i+1}$ resides to the left or right of the directional line-segment between ep_i and ep_{i+1}.

With these parameters established, we map the stroke segment S to a descriptor $d \in D$ using the guidelines outlined in Table 2. To provide a clearer illustration, please refer to Figure 4, which presents an example of a stroke with segments mapped to D [27].
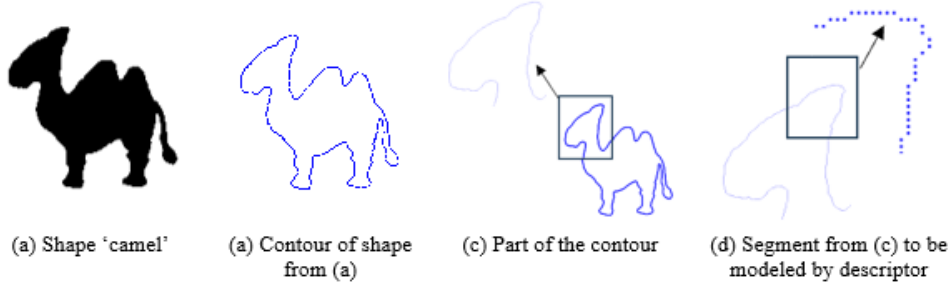

(a) Shape 'camel' (a) Contour of shape from (a) (c) Part of the contour (d) Segment from (c) to be modeled by descriptor

Figure 2. Illustration of the process of modeling shape segment with descriptor

Note: Images taken from widely used MPEG7_CE-Shape-1_Part_B database [26].
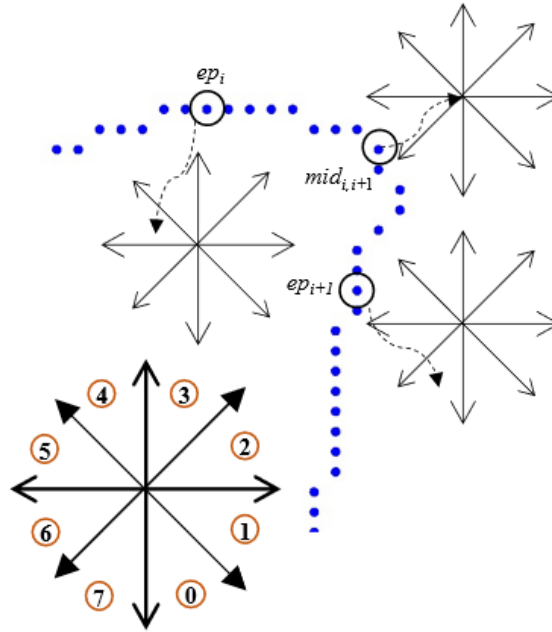
Figure 3. Illustration of space subdivision to identify the type of descriptor to which a segment match

Note: The segment matched is the one from subgraph (d) of Figure 2.

Table 2. Fitting criteria for the three rotation-invariant shape descriptors [27]

Descriptor	Symbol	Fitting Criteria
straight_line	S	pos $(ep_{i+1}) = $ pos $(ep_i) + 4, \mod 8$
cw_curve	C	$\neg S \wedge loc(mid_{i,i+1}) = 'left'$
ccw_curve	V	$\neg S \wedge loc(mid_{i,i+1}) = 'right'$

3.2 Point Suppression

In this section, we discuss the concept of point-suppression and how point-suppression is used in this work. Our discussion of point-suppression is more general than the concept of collinear–points suppression used in

studies [3, 5, 10]. Collinear-points suppression was used successfully for smoothing the contour (boundary of a curve) in 2-dimension [28]. We will used collinear-points suppression as post-processing phase in our dominant point detection method, as discussed later. The Idea of point-suppression is to decide whether a point P on the shape contour C should be retained as a dominant point or not. For this purpose, we need to define a significance value for P. This significance value of P can be defined in numerous ways. In this work, we define this significance value of P in two different ways, that lead to two different ways of point suppression:

- Point suppression by distance thresholding and
- Point suppression by descriptor modeling.

Point suppression by distance thresholding. Let P_j, P_{j+1}, and P_{j+2} be three points on C in sequence (not necessarily consecutive). In point suppression by distance thresholding, P_{j+1} is suppressed if the distance δ from P_{j+1} to the line joining P_j and P_{j+2} is less than some predefined threshold τ. Several researchers have used the perpendicular distance from P_{j+1} to the line joining P_j and P_{j+2} as the distance δ [5, 10]. Parvez and Mahmoud [3] showed that using δ as the minimum distance from P_{j+1} to the line-segment joining P_j and P_{j+2} can have superior performance in preserving shape of the curve C.

Figure 5 illustrates the concept of points suppression by distance thresholding. Here, δ is taken as the perpendicular distance from P_{j+1} to the line joining P_j and P_{j+2}. The point P_{j+1} will be removed as the distance δ is greater than the threshold τ. The new piecewise linear curve C' resulted due this suppression will have one segment less than C.
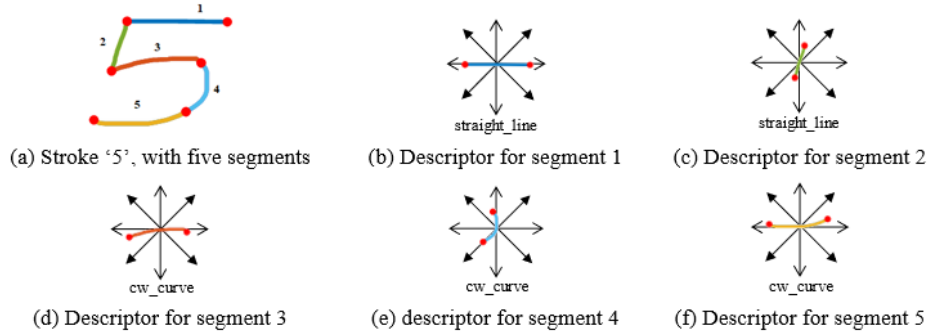

(a) Stroke '5', with five segments   (b) Descriptor for segment 1   (c) Descriptor for segment 2

(d) Descriptor for segment 3   (e) descriptor for segment 4   (f) Descriptor for segment 5

Figure 4. A stroke composed of five segments, with each segment corresponding to an invariant descriptor

(a) Distance threshold   (b) A sample curve   (c) Smoothed curve from (b) using point suppression (τ=1.0.)
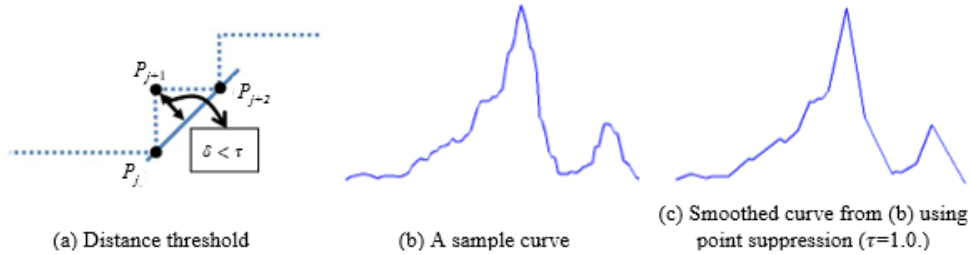
Figure 5. Illustration of points suppression by distance threshold

This concept of suppression of points can be applied to all the points on C and thus the 'redundant' points on C can be deleted. The number of segments and the shape of the resulting curve C' will depend on the threshold τ and the order in which the points on C are considered in the removal process.

There are several ways to determine the value of the threshold τ. Some researchers have used a fixed value for τ [10], others have determined the value of τ adaptively [3, 5, 22]. These adaptive schemes use some optimization techniques to compute the threshold that optimize some performance measures (like weighted sum square error). These schemes are more robust and can be applied to curves of different dimensions and resolutions.

The following procedure *SuppressionByThresholding* summarizes the concept of point suppression by distance thresholding. The results of applying *SuppressionByThresholding* are illustrated in subgraphs (b) and (c) of Figure 5.

        Procedure *SuppressionByThresholding*
            Input: $P = \{(x_i, y_i)\}, i = 1, 2, \ldots, n$, a sequence of n points on contour C.
                A threshold τ.
            Output: P', a subset of P.
            Start

$$P' \leftarrow P$$

Iterate until points cannot be suppressed anymore

For each point p_i from P'

$\circ \rightarrow$ Suppress p_i if $\delta < \tau$.

$\cdot \rightarrow P' \leftarrow P' - \{p_i\}$

End

Point suppression by descriptor modeling. Here, we also consider three points on C in sequence: P_j, P_{j+1}, and P_{j+2}. However, instead of measuring the distance from P_{j+1} to the line segment connecting P_j and P_{j+2}, we consider the shape formed at point P_{j+1}. For this purpose, refer again to Figure 3. If we map P_j, P_{j+1}, and P_{j+2} to $ep_i, mid_{i,i+1}$, and ep_{i+1} respectively, then the curve segment bounded by P_j and P_{j+2} can be mapped to one of the descriptors from Table 2. In point suppression by descriptor modeling, point P_{j+1} will be suppressed if the curve segment bounded by P_j and P_{j+2} is mapped to the descriptor 'straight_line'.

It should be noted here that, to suppress points by descriptors, we need to bound a segment by end points (that is, P_j and P_{j+2} need to be fixed). Since the mapping of the segment to a descriptor can vary based on the locations of P_j and P_{j+2}, we use the following iterative process to suppress points using descriptors. For that purpose, let us define width of a segment S of C be the number of points in S. For example, the width of the segment in Figure 3 is 17. Based on this definition of width of S, the following *SuppressionByDescriptors* procedure captures the process of point suppression using descriptors discussed earlier.

Procedure *SuppressionByDescriptors*

Input: $P = \{(x_i, y_i)\}, i = 1, 2, \ldots, n$, the sequence of n points on contour C.

Set of descriptors from Table 2

Output: P', a subset of P.

Start

$$P' \leftarrow P$$

Iterate until points cannot be suppressed anymore

For $w = 1$ to maxwidth $//$ max Width to be determined experimentally

For each point p_i from P'

$\circ \rightarrow$ Consider the segment S of width $= w + 1$, where p_i is the middle point of S

$\circ \rightarrow$ if S is mapped to straight_line.

$- \rightarrow$ Suppress p_i

$P' \leftarrow P$

End For

End For

End

3.3 Dominant Point Detection

Equipped with the invariant descriptors and the concept of point suppression, we are now in a position to develop a fast and effective dominant point detection algorithm. The proposed algorithm is illustrated with a block diagram (along with an example) in Figure 6. In the following discussion, we delve into the details of each block represented in Figure 6.

Once the contour C of a planar shape is extracted (Block-1), contour C is passed to *SuppressionByDescriptors* procedure to extract the initial set of dominant points (Block-2). This phase of *SuppressionByDescriptors* is run in uniform division mode. This means, when we consider three points on C in sequence $< P_j, P_{j+1}$, and $P_{j+2} >$, the number of points between P_j and P_{j+1} and number of points between P_{j+1}, P_{j+2} is same (that is, P_{j+1} falls exactly at the middle of the segment being mapped to a descriptor). These initial set of dominant points from Block-2 are then suppressed by *SuppressionByDescriptors* procedure with a fixed threshold of 1 (Block-3). This phase of suppression of points with fixed threshold is needed to suppress any collinear points that may remain from the previous phase (Block-2). Using a fixed threshold at this phase avoid estimating the optimal threshold values, unlike other methods [3].

In the next phase (Block-4), the current set of dominant points once again go through *SuppressionByDescriptors* procedure, but in non-uniform division mode. This mode differs from the previous phase (Block-2) of uniform division in a number of ways. Again, suppose we consider three points on C in sequence $< P_j, P_{j+1}$, and $P_{j+2} >$. In non-uniform division, all points from P_j, P_{j+1}, and P_{j+2} are from the set of dominant points retained by Block-3. In addition, the number of points between P_j and P_{j+1} and number of points between P_{j+1}, P_{j+2} is not necessarily the same (that is, P_{j+1} doesn't necessarily fall at the middle of the segment being mapped to a descriptor). Moreover, in the segment defined by P_j and P_{j+2}, P_{j+1} is the only other dominant point retained by Block-3.

Quality measures of the approximations produced by our method can be enhanced by an optional efficient optimization phase. This optimization is called shift optimization. Shift optimization is a simple optimization process where we try to move a dominant point in its neighborhood along contour C, such that some quality measures (discussed in the next section) are optimized. Experimental results show that shift optimization phase adds very little overhead to the running time, compared to the significant improvements obtained by the optimization.
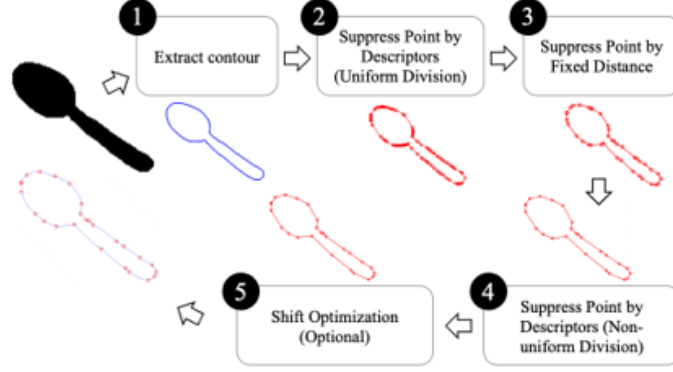

Figure 6. Block diagram of dominant point detection algorithm with an illustrated example

Notes: The original image ('spoon') with 518 contour points is approximated with 26 dominant points.

4 Experimental Results

The proposed method has been extensively evaluated to estimate the quality of approximations and the efficiency of the algorithm. The results have proven to be very promising, especially considering the significant reduction in running time for the MPEG7_CE-Shape-1_Part_B database [24]. In the subsequent sections, we first describe some quality measures commonly used by researchers to estimate the quality of detected dominant points. Following that, we present our experimental results.

4.1 Quality Measures

Evaluating the quality of an approximation is of utmost importance to assess performance, and researchers have introduced various measures for this purpose [6, 9, 10, 22, 29]. These measures are briefly outlined below.

The compression ratio is used to quantify the normalized compression rate of an approximation. It is denoted as $CR = n/n_d$, where n is the number of contour points and n_d is the number of dominant points. However, it doesn't consider the approximation error. An alternative is the integral sum of squared error, $\text{ISE} = \sum_{i=1}^{n} e_i^2$. ISE measures the error of an approximation as the distance e_i from a contour point p_i to the resultant approximating line segment. Nonetheless, ISE can always be minimized by increasing the count of dominant points. This issue with ISE leads us to the use of the weighted sum of square error, $WE = ISE/CR$. Many researchers have adopted this measurement, which merges both the compression ratio and the sum of squared errors [18].

Rosin [20] recognized an imbalance in the two constituents of the WE measure, leading to a bias in favor of approximations with lower ISE (achieved by increasing the number of detected dominant points). This characteristic makes it less suitable for comparing contours with varying numbers of dominant points. Rosin introduced two components, namely fidelity and efficiency. Fidelity evaluates how well the polygonal approximation aligns with the contour in comparison to the optimal polygon in terms of approximation error [20]: Fidelity $= (E_{\text{opt}} / E_{\text{approx}}) \times 100$. Here, E_{approx} represents the error of the polygonal approximation, and E_{opt} signifies the error associated with the optimal algorithm, both adjusted to yield the same number of lines. The optimal polygon achieves the lowest possible error for a given dominant point count.

To achieve a more balanced measure, WE (Weighted Efficiency) has been adjusted by various researchers, resulting in a modified version denoted as $WE_x = ISE/CR^x$, where x is a parameter that regulates the influence of the denominator, thus mitigating the imbalance between the two terms. Common choices for x include 1, 2, and 3 [5, 9, 11, 12]. Carmona-Poyato et al. [5] notably showed that WE_2 outperforms WE. In line with this, we have adopted the WE_x measures to assess the results produced by our algorithm.

To better understand the effects of various measures, consider the example shown in Figure 7. As can be seen in subgraphs (b) and (c) of Figure 7, the ISE is lower in the approximation in subgraph (b) of Figure 7, albeit, with more and sometimes unnecessary dominant points. In contrast, the approximation in in subgraph (c) of Figure 7 uses less number of dominant points, resulting in higher ISE compared to the subgraph (b) of Figure 7. However, the

approximation in subgraph (b) of Figure 7 is preferred over the approximation in subgraph (b) of Figure 7, as the approximation in in subgraph (c) of Figure 7 has lower WE_3 value.
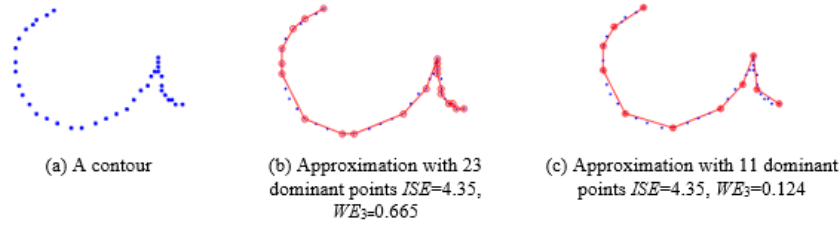

(a) A contour

(b) Approximation with 23 dominant points *ISE*=4.35, *WE₃*=0.665

(c) Approximation with 11 dominant points *ISE*=4.35, *WE₃*=0.124

Figure 7. Illustration of the effects of the number of dominant points on WE_x measures

4.2 Results on Basic Shapes

Table 3. Comparative approximation results for basic shapes

Shape	Method	n_d	CR	ISE	WE	WE_2	WE_3
Chromosome							
(n=60)	Marji and Siy [10]	10	6.00	10.01	1.66	0.277	0.046
	Carmona-Poyato et a1. [24]	11	5.36	9.60	1.79	0.334	0.062
	Masood [11]	15	4.00	3.88	**0.97**	0.243	0.061
	Carmona-Poyato et al. [5]	15	4.00	4.27	1.07	0.267	0.067
	Parvez and Mahmoud [4]	10	6.00	14.34	2.39	0.398	0.066
	Nguyen and Debled-Rennesson [13]	18	3.33	4.06	1.22	0.366	0.110
	Parvez [18]	11	5.46	7.09	1.30	**0.238**	**0.044**
	Parvez [19]	10	6.10	33.54	5.50	0.901	0.148
	Proposed	10	6.10	15.70	2.57	0.422	0.069
	Marji and Siy [10]	17	7.06	28.67	4.06	0.575	0.081
	Carmona-Poyato et al. [24]	17	7.00	37.36	5.33	0.761	0.109
	Masood [11]	23	5.22	9.46	1.81	**0.347**	0.067
	Carmona-Poyato et al. [5]	23	5.22	10.68	2.05	0.391	0.075
Leaf (n=120)	Parvez and Mahmoud [4]	21	5.71	13.82	2.42	0.423	0.074
	Nguyen and Debled-Rennesson [13]	33	3.64	5.56	**1.53**	0.419	0.115
	Parvez [18]	21	5.71	11.98	2.10	0.367	**0.064**
	Parvez [19]	23	5.26	32.50	6.17	1.174	0.223
	Proposed	19	6.36	20.07	3.15	0.495	0.078
Semicircles							
(n=102)	Marji and Siy [10]	15	6.80	22.70	3.34	0.491	0.072
	Carmona-Poyato et al. [24]	11	9.18	59.06	6.03	0.700	0.076
	Masood [11]	26	3.92	4.05	**1.03**	**0.263**	0.067
	Carmona-Poyato et a1. [5]	26	3.92	4.91	1.25	0.319	0.082
	Parvez and Mahmoud [4]	17	6.00	19.02	3.17	0.528	0.088
	Nguyen and Debled-Rennesson [13]	25	4.12	5.42	1.32	0.319	0.078
	Parvez [18]	15	6.80	18.22	2.24	0.329	**0.048**
	Parvez [19]	21	4.91	24.47	4.99	1.017	0.207
	Proposed	19	5.42	20.22	3.73	0.688	0.127
Infinity							
(n=45)	Carmona-Poyato et a1. [24]	9	4.89	7.34	1.50	0.306	0.063
	Masood [11]	11	4.09	2.90	**0.71**	**0.173**	0.042
	Carmona-Poyato et a1. [5]	10	4.50	5.29	1.18	0.261	0.058
	Parvez and Mahmoud [4]	9	5.00	7.35	1.47	0.294	0.059
	Parvez [18]	7	6.43	7.69	1.20	0.186	**0.029**
	Parvez [19]	9	5.00	14.60	2.92	0.584	0.117
	Proposed	11	4.18	4.40	1.05	0.252	0.060

Note: The best results for WEx measures are marked as bold.

Initially, the proposed algorithm is applied to four widely recognized shapes: chromosome, leaf, semicircles, and infinity. These shapes have commonly been used as benchmarks for evaluating dominant point detection algorithms in prior research [3, 5, 7, 10–12, 16, 17, 24]. The results obtained by our algorithm for these four shapes are depicted in Figure 8 [19]. Furthermore, we present a comparative evaluation of the current algorithm against other reported methods in Table 3.
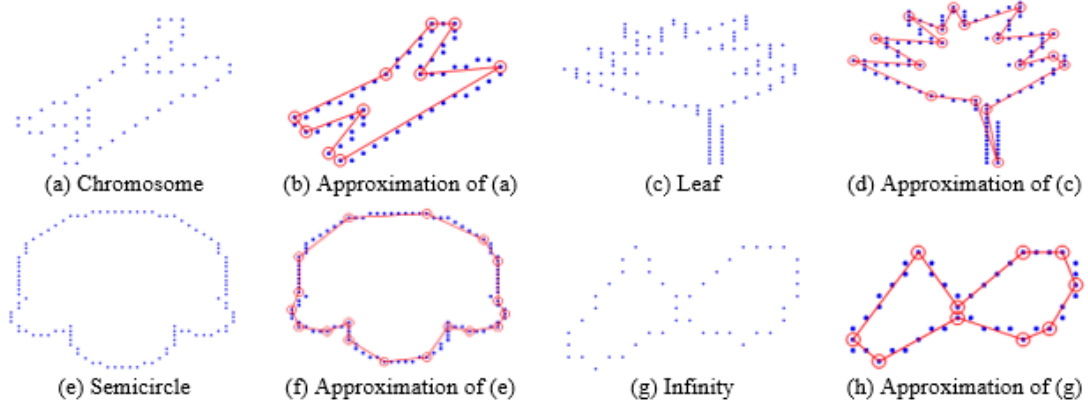

(a) Chromosome (b) Approximation of (a) (c) Leaf (d) Approximation of (c)

(e) Semicircle (f) Approximation of (e) (g) Infinity (h) Approximation of (g)

Figure 8. Polygonal approximations by the proposed method for four popular shapes

As can be seen in Table 3, the outputs from the proposed method are very much comparable with other methods, especially the WE_3 measure. Note that, the proposed method achieves excellent WE_3 measures, although the method only employs a simple shift optimization at the end of the algorithm.

4.3 Results on Large Dataset

While the four basic shapes are simple and provide some insights into the performance of a dominant point detection algorithm, they do not provide extensive information, particularly if we want to evaluate the running time. For this purpose, we experiment with the popular shape database MPEG7_CE-Shape-1_Part_B [26], which is widely used by researchers for testing shape analysis and recognition related works. This dataset contains images with a large number of contour points, making it suitable for testing dominant point detection algorithms. Figure 9 illustrates some outputs of the proposed method for some shapes from the MPEG7_CE-Shape-1_Part_B dataset.

Table 4 shows the statistics of the MPEG7_CE-Shape-1_Part_B dataset and the average number of dominant points for different algorithms for the same dataset. As can be observed from Table 4, our algorithm is comparable to other methods in terms of the average number of dominant points. However, for a more thorough comparison with detailed statistics, please refer to Table 5.

Table 4. Some statistics and approximation results for MPEG7_CE-Shape-1_Part_B database

Dataset	Sample Count	Average # of Contour Points	Average Number of Dominant Points			
			Marji and Siy [10]	Parvez and Mahmoud [3]	Parvez [18]	Proposed
MPEG7_CE-Shape -1 Part B [26]	1402	1272.7	98.6	53.5	81.34	90.61

Table 5 compares the proposed method with three other methods based on four parameters. Table 5 gives us a lot of insights. First, results from other reported methods in Table 5 give us a clue to measure the maxWidth parameter in *SuppressionByDescriptors* procedure. The average CR for other reported methods is around 20, which means, on average, each segment of a shape contour in MPEG7_CE-Shape-1_Part_B dataset is of length of around $n/20$, where n is the number of contour points of a shape. Thus, we set $maxWidth$ to $n/20$. However, to make things general, $maxWidth$ can be always set to $n/2$.

As can be seen in Table 5, the proposed method has very much similar CR compared to other methods. However, the average ISE is slightly higher for the proposed method. This is expected, as the proposed method is not based on optimizations, except that our method uses a simple optimization process as a post-processing step. Higher average ISE of the proposed method leads to higher average WE_3 measure, although the difference with other methods in terms of WE_3 is tolerable.

However, the biggest gain of the proposed method is in terms of running time. All the methods reported in Table 5 have been implemented by MATLAB and run on a MacbookPro (M1-Pro) with 16 GB of RAM. As can be seen in

Table [5], the proposed method runs a minimum of 2.56 times faster compared to other methods. When compared with the most optimal algorithm [18], the proposed method is 13 times faster! This huge gain in running time is due to the simple descriptor-based suppression technique of the proposed method, as opposed to costly operations used in other methods. This indicates that the proposed method can be very suitable for real time applications or for processing shapes with large number of points.

Table 5. Comparisons of the current method with other reported works for MPEG7_CE-Shape-1_Part_B database

Algorithm	Statistics			
	Average CR	Average ISE	Average WE_3	Running Time (sec)
Marji and Siy [10]	20.52	285.41	0.229	112.08
Parvez and Mahmoud [3]	29.24	1783.71	0.154	180.70
Parvez [18]	24.26	280.30	0.126	569.46
Proposed [with all phases]	23.35	2095.59	0.681	43.64
Proposed [w/o phase 4]	17.95	457.39	0.513	43.35
Proposed [w/o phase 5 (optimization)]	23.35	2891.36	0.932	41.47
Proposed [w/o phase 4 and phase 5]	17.95	782.29	0.678	42.27

The proposed method is primarily tailored for offline planar shapes. However, the method discussed in this work can also be applied to online strokes, wherein the algorithm is executed after the user completes the stroke. To illustrate the results of the proposed method for online strokes, please refer to Figure [10]. In Figure [10], we showcase a few examples from the Online-KHATT database [30] for online Arabic handwritten text, along with the results obtained from our algorithm.
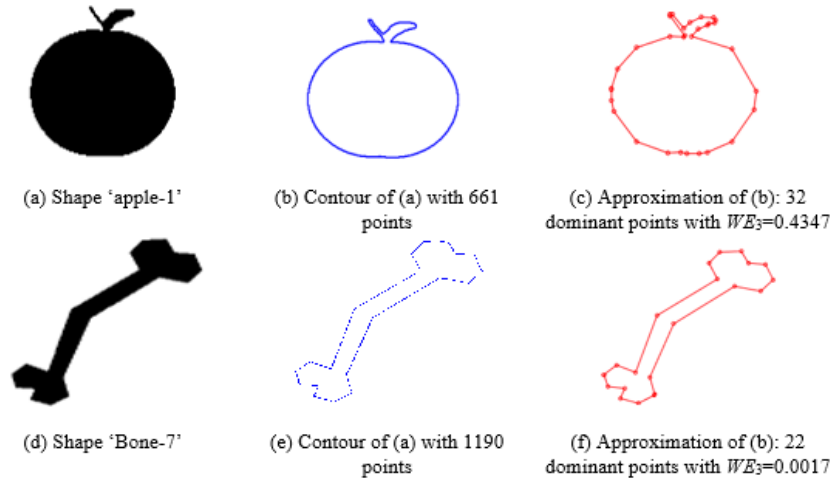

(a) Shape 'apple-1'

(b) Contour of (a) with 661 points

(c) Approximation of (b): 32 dominant points with WE_3=0.4347

(d) Shape 'Bone-7'

(e) Contour of (a) with 1190 points

(f) Approximation of (b): 22 dominant points with WE_3=0.0017

Figure 9. Polygonal approximations by the proposed method for shapes from MPEG7_CE-Shape-1_Part_B dataset

(a) Shape 'ز'

(b) Approximation of (a)

(c) Main body of shape 'ذ'
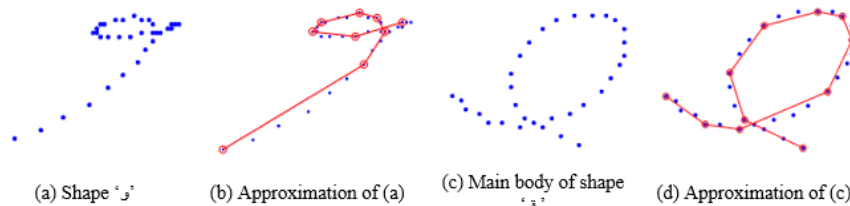
(d) Approximation of (c)

Figure 10. Application of our algorithm for online shapes (data taken from Online-KHATT database [30])

5 Conclusions

In this study, we introduce a novel dominant point detection algorithm based on descriptors. Our innovative algorithm evaluates each point along a shape contour, assigning it a set of invariant descriptors relative to its neighboring points. These descriptors dynamically determine whether a given point qualifies as a dominant point. Notably, our approach circumvents the need for resource-intensive computations typically required to classify a point as a potential dominant point. In addition, our technique demonstrates remarkable speed when compared to other existing methodologies, owing to its reliance on integer operations and its independence from optimization phases. However, the approximations resulting from the current method exhibit higher ISE, as no optimizations are performed while producing the approximations. The presented method could be improved by investigating strategies to reduce the ISE of the estimated approximations.

Funding

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, no. 3, pp. 183–193, 1954. https://doi.org/10.1037/h0054663

[2] D. S. Zhang and G. J. Lu, "Review of shape representation and description techniques," *Pattern Recogn.*, vol. 37, no. 1, pp. 1–19, 2004. https://doi.org/10.1016/j.patcog.2003.07.008

[3] M. T. Parvez and S. A. Mahmoud, "Polygonal approximation of digital planar curves through adaptive optimizations," *Pattern Recogn. Lett.*, vol. 31, no. 13, pp. 1997–2005, 2010. https://doi.org/10.1016/j.patrec.2010.06.007

[4] M. T. Parvez and S. A. Mahmoud, "Arabic handwriting recognition using structural and syntactic pattern attributes," *Pattern Recogn.*, vol. 46, no. 1, pp. 141–154, 2013. https://doi.org/10.1016/j.patcog.2012.07.012

[5] A. Carmona-Poyato, F. J. Madrid-Cuevas, R. Medina-Carnicer, and R. Muñoz-Salinas, "Polygonal approximation of digital planar curves through break point suppression," *Pattern Recogn.*, vol. 43, no. 1, pp. 14–25, 2010. http://doi.org/10.1016/j.patcog.2009.06.010

[6] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, no. 6, p. 284, 1961. http://doi.org/10.1145/366573.366611

[7] C. P. Chau and W. C. Siu, "New nonparametric dominant point detection algorithm," *IEE Proc.-Vis. Image Signal Process.*, vol. 148, no. 5, pp. 363–374, 2001. http://doi.org/10.1049/ip-vis:20010576

[8] A. Kolesnikov, "ISE-bounded polygonal approximation of digital curves," *Pattern Recogn. Lett.*, vol. 33, no. 10, pp. 1329–1337, 2012. http://doi.org/10.1016/j.patrec.2012.02.015

[9] M. Marji and P. Siy, "A new algorithm for dominant points detection and polygonization of digital curves," *Pattern Recogn.*, vol. 36, no. 10, pp. 2239–2251, 2003. http://doi.org/10.1016/S0031-3203(03)00119-5

[10] M. Marji and P. Siy, "Polygonal representation of digital planar curves through dominant point detection—A nonparametric algorithm," *Pattern Recognit.*, vol. 37, no. 11, pp. 2113–2130, 2004. http://doi.org/10.1016/j.patcog.2004.03.004

[11] A. Masood, "Optimized polygonal approximation by dominant point deletion," *Pattern Recognit.*, vol. 41, no. 1, pp. 227–239, 2008. http://doi.org/10.1016/j.patcog.2007.05.021

[12] U. Montanari, "A note on minimal length polygonal approximation to a digitized contour," *Commun. ACM*, vol. 13, no. 1, pp. 41–47, 1970. http://doi.org/10.1145/361953.361967

[13] T. P. Nguyen and I. Debled-Rennesson, "A discrete geometry approach for dominant point detection," *Pattern Recognit.*, vol. 44, no. 1, pp. 32–44, 2011. http://doi.org/10.1016/j.patcog.2010.06.022

[14] D. K. Prasad, M. K. Leung, C. Quek, and S. Y. Cho, "A novel framework for making dominant point detection methods non-parametric," *Image Vis. Comput.*, vol. 30, no. 11, pp. 843–859, 2012. http://doi.org/10.1016/j.imavis.2012.06.010

[15] H. Stone, "Approximation of curves by line segments," *Math. Comput.*, vol. 15, no. 73, pp. 40–47, 1961. http://doi.org/10.1090/S0025-5718-1961-0119390-6

[16] C. H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 8, pp. 859–872, 1989. http://doi.org/10.1109/34.31447

[17] W. Y. Wu, "An adaptive method for detecting dominant points," *Pattern Recognit.*, vol. 36, no. 10, pp. 2231–2237, 2003. http://doi.org/10.1016/S0031-3203(03)00087-6

[18] M. T. Parvez, "Optimized polygonal approximations through vertex relocations in contour neighborhoods," *Image Vis. Comput.*, vol. 34, pp. 1–10, 2015. http://doi.org/10.1016/j.imavis.2014.10.012

[19] M. T. Parvez, "Fast progressive polygonal approximations for online strokes," *Graph. Models*, vol. 129, p. 101200, 2023. http://doi.org/10.1016/j.gmod.2023.101200

[20] P. L. Rosin, "Techniques for assessing polygonal approximations of curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 659–666, 1997. http://doi.org/10.1109/34.601253

[21] P. I. Hosur and K. K. Ma, "A novel scheme for progressive polygon approximation of shape contours," in *1999 IEEE Third Workshop on Multimedia Signal Processing (Cat. No. 99TH8451), Copenhagen, Denmark*. IEEE, 1999, pp. 309–314. http://doi.org/10.1109/MMSP.1999.793854

[22] J. C. Perez and E. Vidal, "Optimum polygonal approximation of digitized curves," *Pattern Recognit. Lett.*, vol. 15, no. 8, pp. 743–750, 1994. http://doi.org/10.1016/0167-8655(94)90002-7

[23] Y. Zhu and L. D. Seneviratne, "Optimal polygonal approximation of digitised curves," *IEE Proc. Vis. Image Signal Process.*, vol. 144, no. 1, pp. 8–14, 1997. http://doi.org/10.1049/ip-vis:19970985

[24] A. Carmona-Poyato, N. L. Fernández-García, R. Medina-Carnicer, and F. J. Madrid-Cuevas, "Dominant point detection: A new proposal," *Image Vis. Comput.*, vol. 23, no. 13, pp. 1226–1236, 2005. http://doi.org/10.1016/j.imavis.2005.07.025

[25] M. Awrangjeb, G. Lu, and C. S. Fraser, "Performance comparisons of contour-based corner detectors," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4167–4179, 2012. http://doi.org/10.1109/TIP.2012.2200493

[26] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002. http://dx.doi.org/10.1109/34.993558

[27] M. T. Parvez, "Invariant shape descriptors for handwritten strokes," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia*, 2020, pp. 1–4. http://doi.org/10.1109/ICCIS49240.2020.9257603

[28] M. T. Parvez, "Arabic handwritten text recognition using structural and syntactic pattern attributes," Ph.D. dissertation, King Fahd University of Petroleum and Minerals, 2010.

[29] N. L. Fernandez-García, L. D. M. Martinez, A. Carmona-Poyato, F. J. Madrid-Cuevas, and R. Medina-Carnicer, "Assessing polygonal approximations: A new measurement and a comparative study," *Pattern Recognit.*, vol. 138, p. 109396, 2023. http://doi.org/10.1016/j.patcog.2023.109396

[30] S. A. Mahmoud, H. Luqman, B. M. Al-Helali, G. BinMakhashen, and M. T. Parvez, "Online-khatt: An open-vocabulary database for arabic online-text processing," *Open Cybern. Syst. J.*, vol. 12, no. 1, 2018. http://doi.org/10.2174/1874110X01812010042