A fast iterative method for dominant points detection of digital curves

CECILIA DI RUBERTO
University of Cagliari
Dept. of Mathematics and Comp. Science
Via Ospedale, 72 09124 Cagliari
ITALY
dirubert@unica.it

ANDREA MORGERA
University of Cagliari
Dept. of Mathematics and Comp. Science
Via Ospedale, 72 09124 Cagliari
ITALY
andrea.morgera@unica.it

Abstract: A new algorithm for detecting dominant points of digitized closed curves is presented. At first a process of localization of maximum curvature points is done. The method adds iteratively contour points as dominant points by a criterion that considers the distance of contour points respect to the set of dominant points calculated in previous steps. It follows a final process of refinement. The performance of our method are successfully compared with other dominant point extraction techniques. Finally, some examples of polygonal approximation are showed.

Key–Words: Curvature, Digital Curve, Dominant Points, Polygonal Approximation.

1 Introduction

In the field of machine vision applications, the aim of shape recognition is to identify an object correctly. There are two typical approaches to the problem: global or internal methods, which use all the points of the objects and contour or external methods which are focused on the analysis of contour points of the objects. Dominant point detection is a very important aspect in contour methods since information on the shape of a curve is concentrated at dominant points having high curvature. This representation simplifies the analysis of the images by providing a set of feature points containing almost complete information of a contour. One of the main advantages of a good representation of 2D shapes through dominant points is the high data reduction and its immediately efficiency in feature extraction and shape matching algorithms. It is well known also that these points play a dominant role in shape perception by humans.

The problem of detecting points of high curvature in 2D shapes has been researched since the early 1970’s [1]-[10]. Most of these algorithms require one or more parameters that specify (directly or indirectly) the region of support in order to measure the local properties at each point of the curve. Some algorithms are not robust in noisy contours, due to fact that the local maximum curvature may be caused by noisy variations on the curve.

In this work, we will focus on an algorithm that attempts to represent shapes with a limited number of dominant points located along their boundaries. We consider a set of points as a dominant set if is possible to reconstruct the contour from these points using some method or interpolation (polygon is the simplest case). The method is based on an iterative approach. The first step is to assign an initial set of points and after detect other possible dominant points, i.e. relevant points, applying an iterative selection based on a particular distance criterion. The results we obtain are then refined by suppressing some overmuch points in critical regions of the shape contour. The detection procedure is compared with methods proposed by [5], [9] and by [10]. These methods relies on the curvature estimation approach and on the determination of the region of support. In section 2 the process to locate initial points set for the iterative method is explained. Section 3 describes the algorithm with its main steps. In the section 4 a refinement technique for a better localization of dominant points is proposed. The experimental results are shown in section 5 and finally the main conclusions are summarized in section 6.

2 Initial point setup

In order to reach a complete set of dominant points of a given a contour, we need of a initial group of starting points. The four vertex theorem [11] states that a simple closed curve in the plane, other than a circle, must have at least four “vertices”, that is, at least four points where the curvature has a local maximum or a local minimum. By the result of this theorem we choose to locate four initial points by searching for local maxima and minima in the signature of the shape. At first
3 Dominant point detection

After the initial points setup the method builds a set of dominant points in the following way:

1) Let \( D = \{ d_i \} \), \( i = 1, \ldots, m \) the initial dominant points set. In the first iteration \( m = 4 \) according to the starting points setup. For each pair of points \( d_i, d_{i+1}, \ i = 1, 2, \ldots, m \) where \( d_{i+1} \) is a neighbor of \( d_i \) (modulo \( m \)) it is possible to sum the distance from each contour point \( p_j \) \( (j = 1, \ldots, n) \) between \( d_i \) and \( d_{i+1} \) and the line that connects the two points:

\[
\text{dist}_j = \frac{|(x_{i+1} - x_i)(y_i - y_j) - (x_i - x_j)(y_{i+1} - y_j)|}{\sqrt{((x_{i+1} - x_i)^2 + (y_{i+1} - y_j)^2}}} \quad (3)
\]

where \((x_i, y_i), (x_{i+1}, y_{i+1})\) and \((x_j, y_j)\) are coordinates of \( d_{i+1}, d_i \) and \( p_j \) respectively. A global distance is calculated as:

\[
T = \sum_j \text{dist}_j \quad , \quad j = 1, \ldots, n \quad (4)
\]

2) In order to find dominant points the value \( T \) of the previous step is compared to a threshold \( s \) defined as \( s = p \times l \), where \( p \) is an input parameter of the algorithm and \( l \) is the number of points of the line that connects \( d_i \) and \( d_{i+1} \). If \( T > s \) we add an intermediate point between \( d_i \) and \( d_{i+1} \) looking for the point with maximum distance to the chord. The current set of points and the new dominant points added are the set of points for the next iteration.

3) The algorithm stops when, after two following iterations, the number of the current set of points is equal to the number of the previous set, i.e. each distance \( T \) is less or equal to the corresponding threshold \( s \).

4 Refinement

Given a dominant points set \( D \) we adopt a refinement technique to suppress dominant point which are near enough between them. Let \( l_{i,j} \) the distance between \( d_i \) and \( d_j \) and \( \tau \) a given threshold, if \( l_{i,j} < \tau \) then we substitute \( d_i \equiv (x_i, y_i) \) and \( d_{i+1} \equiv (x_{i+1}, y_{i+1}) \) with their mid-point :

\[
\overline{d_{i}} \equiv (\overline{x}_k, \overline{y}_k) \quad (5)
\]

where \( \overline{x}_k = \frac{x_i + x_{i+1}}{2} \) and \( \overline{y}_k = \frac{y_i + y_{i+1}}{2} \). A reasonable value of \( \tau = 0.007 \) is set as default.

5 Experimental results

In order to test the effectiveness of our method we perform some experiments, both on typical shapes (introduced by [1] and [5] and commonly used in many studies) and on biological and common tools shapes. In particular we test four contour commonly used curves : chromosome, infinity, leaf and semicircles. Their contour and the set of dominant points, obtained by each method and highlighted with a small
Figure 2: Chromosome shape (50 points): (a) Teh-Chin (15 points), (b) Wu BV (16 points), (c) Wu DYN (17 points), (d) our method (15 points), (e) our method (16 points), (f) our method (17 points)

Figure 3: Infinite shape (45 points): (a) Teh-Chin (13 points), (b) Wu BV (13 points), (c) Wu DYN (13 points), (d) our method (13 points)
Figure 4: Leaf shape (120 points) : (a) Teh-Chin (29 points), (b) Wu BV (24 points), (c) Wu DYN (23 points), (d) our method (23 points), (e) our method (24 points), (f) our method (29 points)

Figure 5: Semicircle shape (102 points) : (a) Teh-Chin (22 points), (b) Wu BV (26 points), (c) Wu DYN (27 points), (d) our method (23 points), (e) our method (27 points)
Table 1: Comparative results for the proposed method

<table>
<thead>
<tr>
<th>Contour</th>
<th>Method</th>
<th>m</th>
<th>CR</th>
<th>$E_2$</th>
<th>$s$</th>
<th>$E_2/CR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome (points=60)</td>
<td>Our method</td>
<td>15</td>
<td>4</td>
<td>6.5</td>
<td>.5</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>16</td>
<td>3.75</td>
<td>4.62</td>
<td>.475</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>17</td>
<td>3.53</td>
<td>3.74</td>
<td>.45</td>
<td>1.06</td>
</tr>
<tr>
<td></td>
<td>Teh-Chin</td>
<td>15</td>
<td>4</td>
<td>7.19</td>
<td>none</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>Wu$_{bv}$</td>
<td>16</td>
<td>3.75</td>
<td>4.70</td>
<td>n.a.</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>Wu$_{dyn}$</td>
<td>17</td>
<td>3.53</td>
<td>5.01</td>
<td>n.a.</td>
<td>1.42</td>
</tr>
<tr>
<td>Leaf (points=120)</td>
<td>Our method</td>
<td>23</td>
<td>5.22</td>
<td>14.32</td>
<td>.45</td>
<td>2.74</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>24</td>
<td>4.75</td>
<td>12.73</td>
<td>.42</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>25</td>
<td>4.14</td>
<td>7.90</td>
<td>.356</td>
<td>1.91</td>
</tr>
<tr>
<td></td>
<td>Teh-Chin</td>
<td>29</td>
<td>4.14</td>
<td>14.96</td>
<td>none</td>
<td>3.62</td>
</tr>
<tr>
<td></td>
<td>Wu$_{bv}$</td>
<td>24</td>
<td>5</td>
<td>17.40</td>
<td>n.a.</td>
<td>3.48</td>
</tr>
<tr>
<td></td>
<td>Wu$_{dyn}$</td>
<td>23</td>
<td>5.22</td>
<td>22.44</td>
<td>n.a.</td>
<td>4.3</td>
</tr>
<tr>
<td>Semicircle (points=102)</td>
<td>Our method</td>
<td>23</td>
<td>3.53</td>
<td>4.43</td>
<td>.4487</td>
<td>2.37</td>
</tr>
<tr>
<td></td>
<td>Our method</td>
<td>27</td>
<td>3.78</td>
<td>11.65</td>
<td>.44</td>
<td>1.51</td>
</tr>
<tr>
<td></td>
<td>Teh-Chin</td>
<td>22</td>
<td>4.64</td>
<td>6.85</td>
<td>.44</td>
<td>1.51</td>
</tr>
<tr>
<td></td>
<td>Wu$_{bv}$</td>
<td>26</td>
<td>3.92</td>
<td>9.04</td>
<td>n.a.</td>
<td>2.18</td>
</tr>
<tr>
<td></td>
<td>Wu$_{dyn}$</td>
<td>27</td>
<td>3.78</td>
<td>9.92</td>
<td>n.a.</td>
<td>2.49</td>
</tr>
<tr>
<td>Infinite (points=45)</td>
<td>Our method</td>
<td>13</td>
<td>3.46</td>
<td>3.65</td>
<td>.4</td>
<td>1.05</td>
</tr>
<tr>
<td></td>
<td>Teh-Chin</td>
<td>13</td>
<td>3.46</td>
<td>5.93</td>
<td>none</td>
<td>1.71</td>
</tr>
<tr>
<td></td>
<td>Wu$_{bv}$</td>
<td>13</td>
<td>3.46</td>
<td>5.40</td>
<td>n.a.</td>
<td>1.56</td>
</tr>
<tr>
<td></td>
<td>Wu$_{dyn}$</td>
<td>13</td>
<td>3.46</td>
<td>5.60</td>
<td>n.a.</td>
<td>1.62</td>
</tr>
</tbody>
</table>

The polygonal approximation for some real images like Africa’s map and maple leaf (taken by [12]) are shown in fig.7 and 8, respectively. We work with Matlab 2008 running on a Linux 64 bit operating system running on AMD64 6000. The Africa’s map con-

circle, are shown in figs.2-5. Our method is compared with others by showing results with the same or nearest number of dominant points.

The proposed method is compared with the other methods and results are shown in table 1. We report some meaningful measures for each tested method: the number of dominant points $m$, the compression ratio (CR), the $E_2$ error and the $E_2/CR$ ratio. CR is defined as the ratio between the number of dominant points $m$ and the number of contour points $n$. The $E_2$ measure is typically used to evaluate the effectiveness of polygonal approximation. It is defined as:

$$E_2 = \sum_{j} d_j, \quad j = 1, \ldots, n$$  \hspace{1cm} (6)$$

where $d_j$ is the distance of a contour point $p_j, j = 1, \ldots, n$ from the segment between $p_i^*$ and $p_{i+1}^*$, $i = 1, \ldots, m$ is the $i$-th of dominant points such that $p_i^* \leq p_i \leq p_{i+1}^*$. We report also the value of threshold $s$ chosen. The results clearly demonstrate better performance of our method respect to other techniques. In fact the $E_2$ error (and subsequently the $E_2/CR$ value) is always lower keeping the same number of dominant points of other methods. Viceversa if we set an acceptable value of $E_2$ error our method gives the lowest number of dominant points, i.e. well-located points, compared to the other.

We test our method and compare it to the other dominant points extraction techniques on common shapes (taken by [13] and [14]) also. In fig. 6 it’s possible to see some shapes we used in experiments. These images are processed by thresholding camera taken objects and are affected by noise. For each image the graph which shows the $E_2$ error trend against the dominant point number is plotted. We use the log scale for the error $E_2$. We obtain such values by varying the input parameter for each method except for the Teh-Chin method which not requires any. These graphs confirm the effectiveness of our method. Worse results of other methods probably are due to the noise sensitivity.
Figure 6: Shapes used in the experiments.
Figure 7: Africa’s map: (a) original shape, (b) 20 points approximation, (c) 30 points approximation, (b) 40 points approximation.

Figure 8: Maple leaf: (a) original shape, (b) 40 points approximation, (c) 45 points approximation, (b) 50 points approximation.
tour is composed by 1364 points: we calculate the approximation with 20, 30 and 40 points respectively (the dotted line is the original contour). The resulting computational time is 0.0756 sec., 0.0731 sec and 0.0659 sec. The maple leaf shape is composed by 1630 points: we calculate the approximation with 40, 45 and 50 points respectively (the dotted line is the original contour). The resulting computational time is 0.1343 sec., 0.1449 sec and 0.1895 sec. respectively.

6 Conclusions

The problem to find a fast and accurate method to find a set of dominant points given a contour shape is addressed in this paper. The proposed method not only has a very low computational time and robustness to noise but also produces a good polygonal approximation while keeping low the $E_2$ error. By empirical evidence we notice that a threshold value in a range around 0.5 give the best $E_2/CR$ value and an improvement of the proposed method by doing without the threshold input parameter in future works is planned.

References:


