Boundary and Shape Complexity of a Digital Object

Mousumi Dutt[1](✉) and Arindam Biswas[2]

[1] Department of Computer Science and Engineering, St. Thomas' College of Engineering and Technology, Kolkata, India
duttmousumi@gmail.com
[2] Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, India
barindam@gmail.com

Abstract. The orthogonal convex hull is the minimal area convex polygon covering a digital object whereas an orthogonal convex skull is the maximal area convex polygon inscribing the digital object. A quantitative approach to analyse the complexity of a given hole-free digital object is presented in this paper. The orthogonal convex hull and an orthogonal convex skull are used together to derive the complexity of an object. The analysis is performed based on the regions added while deriving the orthogonal convex hull and the regions deleted while obtaining an orthogonal convex skull. Another measure for shape complexity using convexity tree derived from the orthogonal convex skull is also presented. The simple and novel approach presented in this paper is useful to derive several shape features of a digital object.

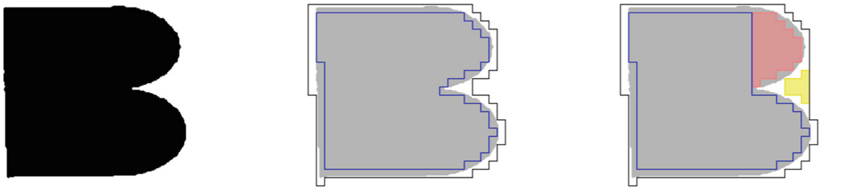
Keywords: Outer isothetic cover · Inner isothetic cover · Orthogonal convex hull · Orthogonal convex skull · Concavity · Convexity · Convexity tree

1 Introduction

Shape analysis of digital objects has various applications in diversifying fields. Similarly, boundary complexity analysis is also important in this context. The detailed patterns of boundary give an idea about the shape of the object. The description of the detailed pattern of the contour can be analysed using wavelet local extrema, which are obtained through wavelet transform [9]. The similarities between two patterns, examined by this method uses the detailed features of contours and their arrangement. From 2D contours of an image, 3D shape can be inferred, which is an important problem in machine vision. Two kinds of symmetries, i.e., parallel and mirror symmetries, give significant information about the surface shape for a variety of objects [15]. Image retrieval can be performed based on color, shape, and spatial properties of an image. Such a technique is proposed in [8], in which a prototype has been implemented to retrieve a particular image

R.P. Barneva et al. (Eds.): CompIMAGE 2016, LNCS 10149, pp. 105–117, 2017.
DOI: 10.1007/978-3-319-54609-4_8

from an image database, and to calculate the symmetry between two images. The boundary analysis can be accomplished using features, like shape and texture, which represents the objects [11].



(a) Object, A (b) Outer and Inner Isothetic Cover, A_{out} and A_{in} (c) OH(A) and OCS(A)

Fig. 1. A sample 2D object and its orthogonal convex hull and skull ($grid\,size = 10$).

The applications of shape analysis and retrieval are not bounded in image processing domain. It has various real-life applications. Shape analysis can be used for road-sign detection in conjunction with image segmentation [10]. Shape based features can also be applied to construct OCR for Bengali script [12]. The working conditions of content-based retrieval, patterns of use, types of pictures, the role of semantics, the sensory gap, and the computational steps for image retrieval systems are presented in [14].

In this paper, measures for the boundary complexity and the shape complexity of a digital object are proposed. The boundary of a digital object has a close relation with outer isothetic cover (orthogonal polygon that tightly covers the digital object) and the inner isothetic cover (orthogonal polygon which tightly inscribes the digital object) [3,4]. Here, the complexity of the boundary of a digital object is analyzed by using the orthogonal hull and skull in conjunction with each other. The boundary is analyzed based on the portions added while deriving the orthogonal convex hull [5,6] and the portions deleted while deriving an orthogonal convex skull [1,7] from inner isothetic cover. The added (discarded) portions imply concave (convex) region in the contour of the object. The shape complexity gives a measure of the global shape of the object.

A digital object has been shown in Fig. 1(a) and its outer isothetic cover (black line), A_{out}, and inner isothetic cover (blue line), A_{in}, are shown Fig. 1(b). The yellow portion is added to outer cover to obtain the orthogonal convex hull (OH(A)) and the red portion is deleted from inner cover to extract an orthogonal convex skull (OCS(A)) (Fig. 1(c)). The resulting skull is shown in blue line and the hull in black line (Fig. 1(c)). The object has a concavity on its contour, which can be identified from one added portion and one deleted portion. These information are useful for analysing the boundary complexity and shape complexity of the digital object.

The paper is organized as follows. Section 2 includes definitions and an overview of constructing orthogonal hull and skull. The boundary complexity and shape complexity are discussed in Sect. 3. A discussion of shape complexity from convexity tree is proposed in Sect. 4. Section 5 contains experimental results along with the data obtained from digital objects which are related to complexity analysis. Concluding remarks are presented in Sect. 6.

2 Definitions and Preliminaries

A (finite) subset of $\mathbb{Z}^2$ in which every pair of points is k-connected[1] is called a *k-connected set*. A *digital object A* is defined to be an 8-connected subset of $\mathbb{Z}^2$ whose complement $\mathbb{Z}^2 \setminus A$ is a 4-connected set [13]. The *background grid* is given by $\mathcal{G} = (\mathcal{H}, \mathcal{V})$, where $\mathcal{H}$ and $\mathcal{V}$ represent two sets of equi-spaced horizontal and vertical grid lines respectively. The *grid size*, g, is defined as the (integer) distance between two consecutive horizontal/vertical grid lines. A *grid point* is the point (with integer coordinates) of intersection of a horizontal and a vertical grid line.

P is said to be an *orthogonal polygon* if and only if each of its vertices is a grid point and each of its edges lie on a grid line. P is an *orthogonal convex polygon* if and only if its intersection with any horizontal or vertical line is either a single line segment or empty. The *orthogonal convex hull*, or simply *orthogonal hull*, of a digital object A, denoted by $OH(A)$, is the smallest area orthogonal polygon such that (i) no point $p \in A$ lies on or outside $OH(A)$ and (ii) intersection of $OH(A)$ with any horizontal or vertical line is either empty or a line segment. An *orthogonal convex skull*, or simply *orthogonal skull*, of a digital object A, denoted by $OCS(A)$, is a maximal-area orthogonal polygon such that (i) no point $p \in \mathbb{Z}^2 \setminus A$ lies on or inside $OCS(A)$ and (ii) $OCS(A)$ is orthogonally convex.

The construction of outer isothetic cover and inner isothetic cover are depicted in Sects. 2.1 and 2.2. The description of the method of deriving orthogonal convex hull is presented in Sect. 2.3 and that of orthogonal convex skull in Sect. 2.4.

2.1 Deriving the Outer Isothetic Cover, (*OIC*)

The outer isothetic cover, A_{out}, is the minimum-area orthogonal polygon that covers the digital object, A, imposed on background grid, $\mathcal{G}$. The algorithm in [3,4] computes the ordered set of vertices of A_{out} using a combinatorial classification of the grid points lying on/inside/outside the object boundary. The characteristics of a grid point p in $\mathcal{G}$ is determined by object containments of the four neighboring cells of size $g \times g$ incident at p. If the number of cells occupied by the object, incident at p is $i \in [0, 4]$, then p is classified to class C_i as shown in Fig. 2. The significance of a class is as follows. (i) C_0: p is not a vertex since none

[1] Two points p and q are said to be k-connected ($k = 4$ or 8) in a set S if and only if there exists a sequence $\langle p = p_0, p_1, \ldots, p_n = q \rangle \subseteq S$ such that $p_i \in N_k(p_{i-1})$ for $1 \leq i \leq n$. The 4-neighborhood of a point (x, y) is given by $N_4(x, y) = \{(x', y') : |x - x'| + |y - y'| = 1\}$ and its 8−neighborhood by $N_8(x, y) = \{(x', y') : \max(|x - x'|, |y - y'|) = 1\}$.

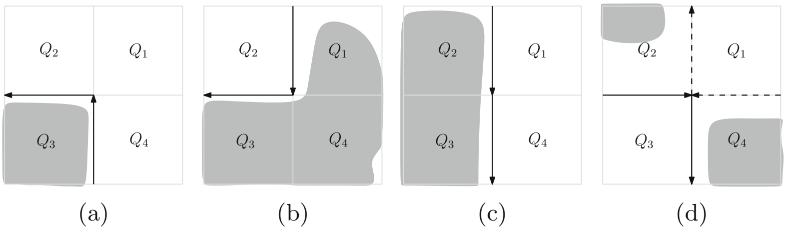


Fig. 2. Different vertex types.

of Q_is has object containment; (ii) C_1: Q_i is a 90° vertex of A_{out} (Fig. 2(a)); (iii) C_2: (a) if two adjacent cell has object containment, then p is an edge point (Fig. 2(c)); (b) if diagonally opposite cells contain object, then p is a 270° vertex of A_{out} (Fig. 2(d)); (iv) C_3: p is classified as a 270° vertex (Fig. 2(b)); (v) C_4: p is not a vertex of A_{out} and lies inside A_{out}.

The start point can be derived from the given top-left point of the digital object. The traversal is made in the anticlockwise direction such that the background lies right of the traversal. During the traversal, from each grid point, v_{i-1}, the next direction, d_i, of traversal is decided by the type, t_i, of the grid point, v_i, and the previous direction, d_{i-1}. The next direction, d_i, is given by $d_i = (d_{i-1} + t_i) \bmod 4$, where $d_i \in [0, 3]$, indicating the direction towards right, top, left, and bottom respectively. Once d_i is computed, the next grid point is determined, and its class is evaluated. Henceforth, in this paper, a 90° vertex is referred as a Type 1 vertex and a 270° vertex as a Type 3 vertex.

2.2 Deriving the Inner Isothetic Cover, (*IIC*)

The inner isothetic cover, namely A_{in}, is the maximum-area orthogonal polygon inscribing the digital object A, which is imposed on the background grid $\mathcal{G}$. The construction of inner isothetic cover is similar to the outer isothetic cover (Sect. 2.1), except for the consideration of the grid point, q. A grid point, q, is classified based on the number of fully occupied cells incident at q.

2.3 Construction of Orthogonal Convex Hull, (*OH*)

The orthogonal hull [5,6] can be detected without any prior knowledge of outer isothetic cover. The concavities in the outer cover are detected and removed when the object boundary is traversed along the grid lines, as mentioned in Sect. 2.1. A region is said to be concave if it has two or more consecutive Type 3 vertices. If the intersection of a horizontal or vertical grid line with the orthogonal polygon has more than one line segment, then it can be said that the object contains one or more concavities. The goal is to identify such regions and derive the edges of the orthogonal hull such that the properties of orthogonal convexity are maintained. In this incremental algorithm, the part of the orthogonal hull obtained upto a point does not contain two consecutive Type 3 vertices which acts as the invariant of the algorithm. Whenever a concavity is detected, necessary combinatorial rules are applied to maintain the algorithm invariant and to ensure the orthogonal convexity, thereof. However, rest of the patterns, **13**, **31**, and **11**, are in conformance with the algorithm invariant and hence do not violate the properties of orthogonal convexity.

2.4 Construction of Orthogonal Convex Skull (*OCS*)

The algorithm to determine an orthogonal convex skull [1,7] first obtains an inner isothetic cover (Sect. 2.2). An orthogonal convex skull is computed by applying rules on the obtained inner isothetic cover. The resulting skull is not unique and it is dependent on the direction of traversal and the starting point of traversal.

Two or more consecutive Type 3 vertices imply a concave region, which defies the properties of orthogonal convexity, as the intersection of a vertical or horizontal grid line with the orthogonal polygon (here, A_{in}) has more than one line segment. The concavity line passing through two consecutive Type 3 vertices, divides the polygon into three different parts: two separate sub-polygons lying on one side of l, and the rest of the polygon at the other side. To achieve orthogonal convexity, one of these two sub-polygons has to be dropped. Hence, first we check whether dropping of any sub-polygon divides (disconnects) the polygon into two parts. If so, then it cannot be dropped. If both the sub-polygons do not affect the connectivity, then the sub-polygon having larger area is included in the skull, thus maximizing the area of the skull. As a result, the convex skull, obtained thereof, also contains no two consecutive vertices of Type 3. Here, in this paper, this algorithm is slightly modified, to obtain OCS for each of the isothetic polygons when the object is disconnected.

3 Boundary Complexity and Shape Complexity

The boundary complexity can be analysed from the boundary of orthogonal convex hull and skull. Orthogonal convex hull is a superset of the outer isothetic cover and orthogonal convex skull is a subset of the inner isothetic cover. Several analyses are presented in this paper using A_{out} with $OH(A)$, A_{in} with $OCS(A)$, A_{out} with A_{in}, and $OCS(A)$ with $OH(A)$.

Let us consider A_{out} and $OH(A)$ together for complexity analysis. The number of portions included in $OH(A)$ provides one measure for the complexity of A. The number of regions added may be zero or one or more. If it is zero, we can certainly say that A_{out} is convex. A shape with more concavities is considered to be more complex. Wherever there is a concavity in the object, a region is included in $OH(A)$ to make it convex. When only one region is added, there are two scenarios depending on the area of the added regions. If the added area is small w.r.t. A_{out}, then the complexity is negligible. If the added area is large w.r.t. A_{out}, then it can be inferred that the complexity lies in the portion of the object where the region has been added. Thus, it can be said that the shape is complex. On the other hand, if the area of the added portion is small enough, then it implies that the object is less complex. If more than one region is added, then there are more than one concavity in A_{out}. Depending on the area of each such portions, the complexity can be determined.

Now considering $OCS(A)$ in conjunction with A_{in}, while determining $OCS(A)$ some convex portions are discarded. Based on the number of portions discarded and the area of those portions, the complexity of the object can be analysed. Similarly, if the number of discarded region is zero, we can certainly say that A_{in} is convex. However, it cannot be inferred that the object is not complex. Because it may happen that the object contains a narrow portion, where A_{in} has not entered. If one region is discarded and the area is large w.r.t. A_{in}, it implies that the complexity lies in that portion of the object. It generally happens with spiral shaped object. On the other hand, if the area is small enough, then it implies that the object is less complex. If the number of discarded portion is greater than one, then there are more than one concavities in A_{in}. Depending on the area of each such portion, the complexity can be determined as said above.

Also, by considering A_{out} and A_{in}, some analyses can be done. If there is a long neck in A, A_{in} may not enter the portion (depending on the grid size and the imposition of the object on the background grid). On the other hand, if the object contains long neck like protrusion in it, A_{out} may not be able to detect it for the same reason. Several measures for boundary complexity and shape complexity are discussed below (Sects. 3.1 and 3.2).

3.1 Boundary Complexity

Three measures for boundary complexity are defined here. Let p, p_i, p_o, p_s, and p_h be the perimeter of A, A_{in}, A_{out}, $OCS(A)$, and $OH(A)$ respectively. The boundary complexity can be defined as follows.

$$\zeta_1 = \frac{(p_h - p_o) + (p_i - p_s)}{p} \times 100 \tag{1}$$

The range is $0 < \zeta_1 < 100$. It is the percentage of the boundary for which the object becomes non-convex. When the object is convex, the value of ζ_1 is equal to zero. High value of ζ_1 implies more concavities (i.e., more complexity) along the boundary. When the grid size decreases, the complexity in the boundary increases, e.g., more concavities are detected along the contour.

During the construction of orthogonal hull, some portions of the outer isothetic cover are included inside it. For the construction of orthogonal convex skull, some portions are removed from the inner isothetic cover. Let $\triangle\ p_o$ be the total perimeter included to construct the orthogonal hull. Similarly, $\triangle\ p_i$ denotes the total perimeter excluded from the inner isothetic cover to construct orthogonal convex skull. The two other measures for boundary complexity can be defined based on these two parameters as follows.

$$\zeta_2 = \frac{\triangle\ p_o}{p_o} + \frac{\triangle\ p_i}{p_i} \tag{2}$$

$$\zeta_3 = \frac{|\frac{\Delta p_o}{p_o} - \frac{\Delta p_i}{p_i}|}{max\{\frac{\Delta p_o}{p_o}, \frac{\Delta p_i}{p_i}\}} \tag{3}$$

ζ_2 is the sum of the complexities in the inner and the outer boundary of the object. The value of ζ_2 is higher in the lower grid sizes compared to the higher grid sizes. The range of ζ_2 is $0 \leqslant \zeta_2 < 2$. When the object is convex then the value of ζ_2 is zero. When the object is too complex most of the portions of outer cover will be included in the hull and most of the portions of the inner cover will be discarded. The value of each fraction will be almost equal to 1. ζ_3 gives a measure on the difference between the inner and the outer complexity. Its range is $0 \leqslant \zeta_3 \leqslant 1$. If the value is zero then the complexity of the inner and outer boundaries are almost same. Otherwise, it means the inner (outer) boundary contains some complexities, which are not detected in the outer (inner) boundary. It is maximum when either inner cover or outer cover is convex. ζ_3 is not dependent on ζ_2.

3.2 Shape Complexity

The two measures for shape complexity are discussed here. Let a, a_i, a_o, a_s, and a_h be the area of A, A_{in}, A_{out}, $OCS(A)$, and $OH(A)$ respectively. The shape complexity can be proposed w.r.t. the area (or regions) of the object which makes the object non-convex. It can be determined as follows.

$$\zeta_4 = \frac{(a_h - a_o) + (a_i - a_s)}{a} \times 100 \tag{4}$$

The range is $0 < \zeta_4 < 100$. This measure determines the percentage of the boundary for which the object becomes non-convex. When the object is convex, the value of ζ_4 is almost equal to zero. High value of ζ_4 implies more complex shape. For lower grid sizes the value of shape complexity increases.

The second measure of the shape complexity can be defined as follows.

$$\zeta_5 = \frac{a_s}{a_h} \tag{5}$$

Its range is $0 < \zeta_5 < 1$. When the object is very complex then $\zeta_5 \simeq 0$ as the area of an orthogonal skull will be much lesser and that of the orthogonal hull will be much higher. The area of the orthogonal convex skull can never be zero, so lower bound of ζ_5 can never be equal to zero. When the object is convex both areas will be almost equal. In case of a convex object, the orthogonal hull is the outer isothetic cover and the orthogonal skull is inner isothetic cover. Always the area of the outer isothetic cover of an object will be greater than that of inner isothetic cover. Thus, the upper bound of ζ_5 cannot be equal to one. If the value of ζ_5 is near to one, it means that the object is less complex. Otherwise, the object is very complex.

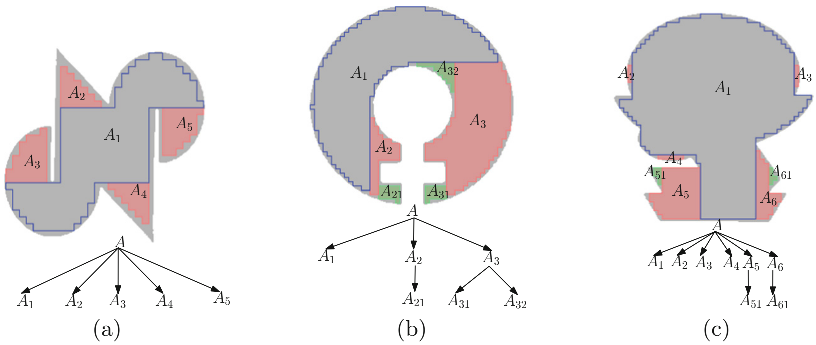


Fig. 3. Convexity tree for three digital objects: (a) `Logo 211` for $g = 8$; (b) `Logo 844` for $g = 7$; (c) `Logo 347` for $g = 5$.

4 Shape Complexity from Orthogonal Convexity Tree

The orthogonal convexity tree can be formed from the orthogonal convex skull. A_{in} consists of the orthogonal convex skull and discarded portions which are actually orthogonal polygons. Some of these are convex and some are not. If orthogonal convex skull can be obtained recursively on the non-convex components till all the components are orthogonally convex, then an orthogonal convexity tree can be determined as shown in Fig. 3. The concept of orthogonal concavity tree is already proposed in [2]. These are useful for determining the shape complexity of a digital object.

Numerous information can be found from the convexity tree, e.g., the depth of convexity tree, the number of children, the number of children per level, etc. If the convexity tree branches out uniformly, then the corresponding object may be symmetric (provided the area of a pair of convex components are same). The object in Fig. 3(a) is symmetric. Here, all the leaves are at same depth. Each pair $(A_2, A_4$ and $A_3, A_5)$ has almost similar area and perimeter. It is to be noted here that area and perimeter comparison are not the only criteria to check whether the object is symmetric or not. If the depth of the tree is higher and there are only one child at each level of the tree, then the shape of the object is very complex and the convexity tree will be unbalanced (e.g., spiral-shaped objects). The depth of the convexity tree is two for the objects shown in Fig. 3(a) and (b). These are more complex objects compared to Fig. 3(a) whose depth is one.

The total number of concavities in the object boundary can be determined from the convexity tree. It is the total number of nodes in the tree (internal nodes and leaves except root) minus one. The number of convex regions which are discarded is the number of concavities in the IIC. All the leaves in the convexity tree correspond to convex regions of the object. If the area is very small w.r.t. the area of the object, then the concavity is insignificant. Let a_x be the area of a convex region. If $\frac{a_x}{a} \simeq 0$, then the corresponding concavity is insignificant. It may not be detected when the object is imposed on higher grid size.

5 Experimental Results

Some of the experimental results are included to explain the complexity analysis in Figs. 4, 5 and 6 and the corresponding data are shown in Table 1. The algorithm of orthogonal hull and skull were implemented in C Ubuntu 14.04. The orthogonal convex skull is shown in blue border whereas orthogonal convex hull with black border. The regions which are included in the orthogonal convex hull are shown in yellow color. The discarded regions to construct orthogonal convex skull are shown in red color. ζ_1 and ζ_2 decreases when grid size increases. For less complex boundary the value is lower. The inner isothetic cover of `Logo 220` for $g = 20$ is convex and its outer isothetic cover has only one concavity. The value of ζ_1 and ζ_2 are very low for this object, whereas $\zeta_3 = 1$ (maximum value).

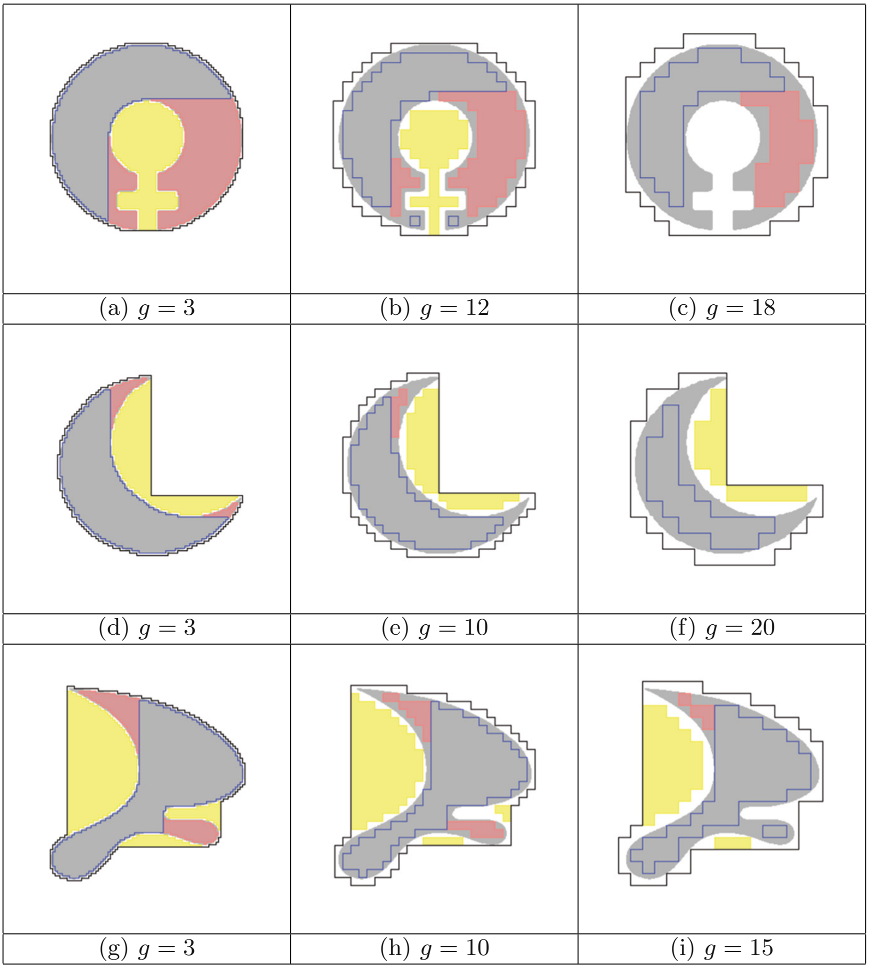


Fig. 4. $OH(A)$ and $OCS(A)$ on a set of objects (`Logo 844`, `Logo 181`, and `Logo 426`) which contains circular portion (g refers to grid size). (Color figure online)

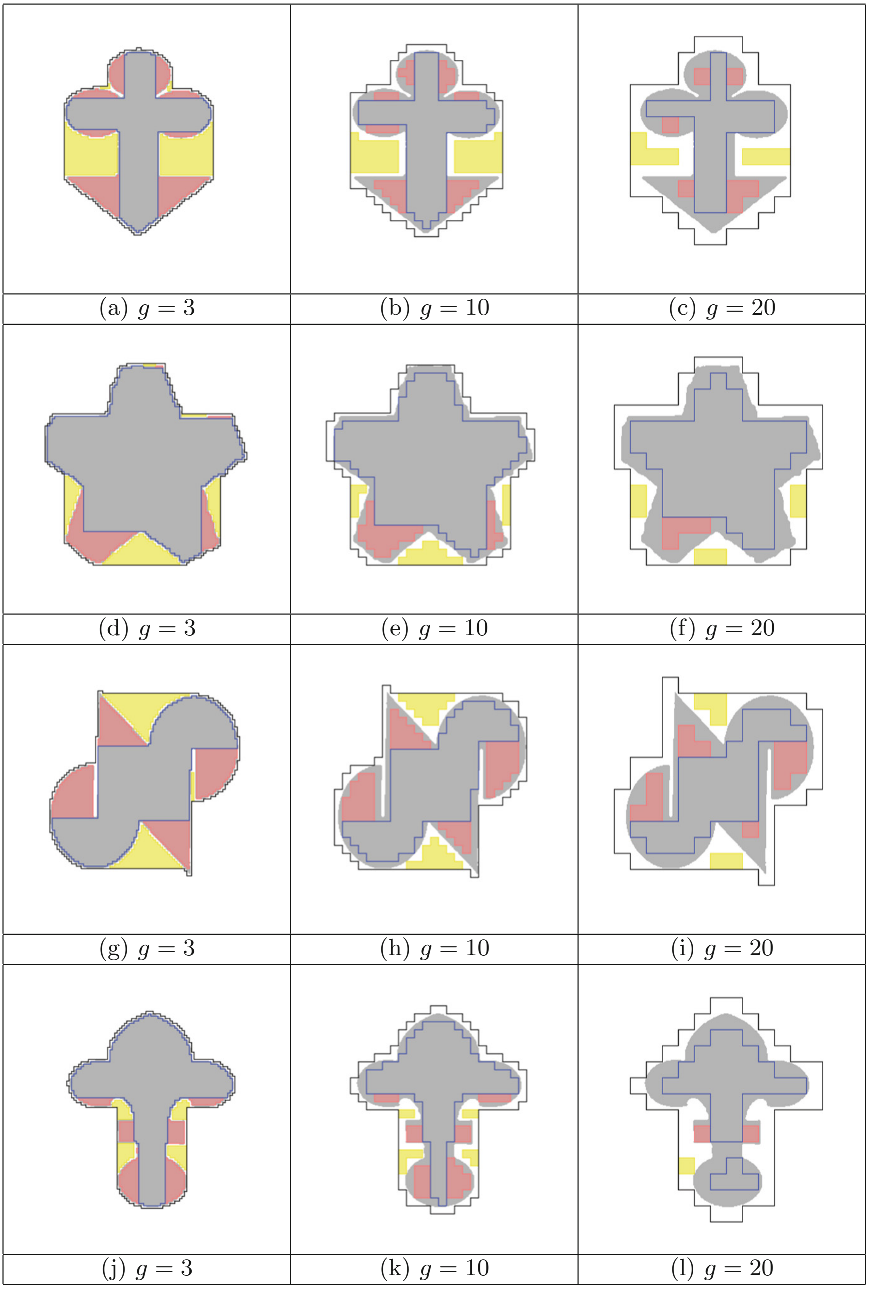
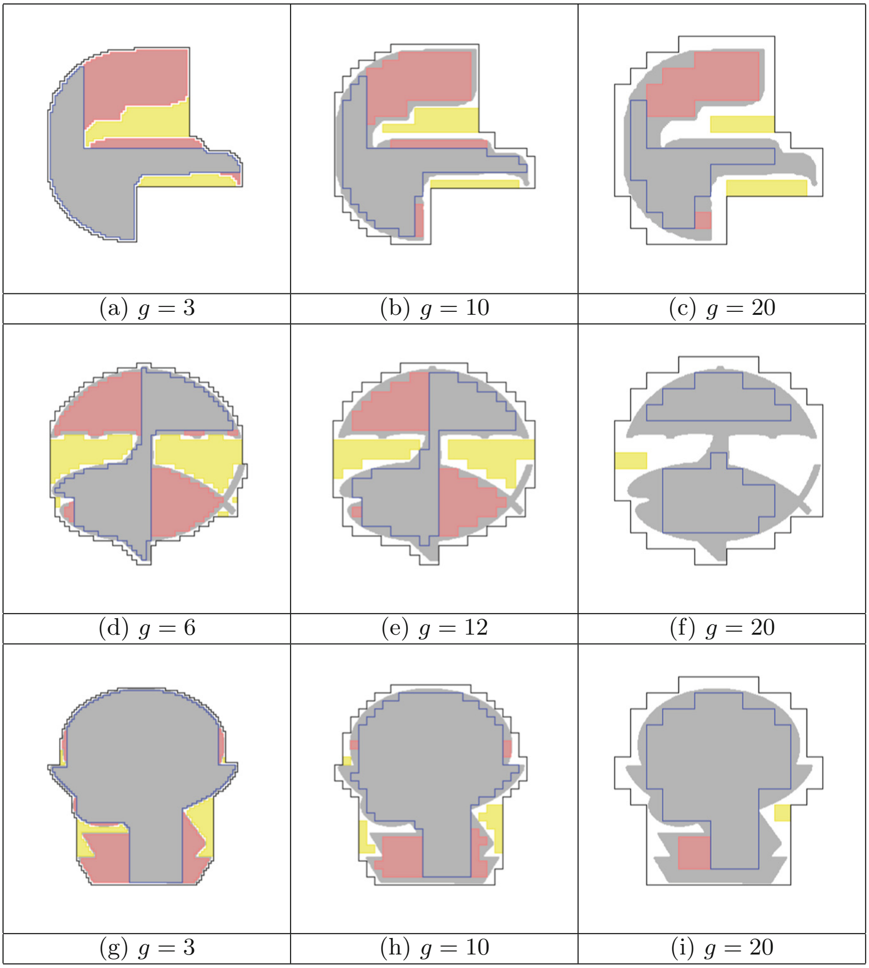


Fig. 5. $OH(A)$ and $OCS(A)$ on a set of symmetric objects (`Logo 1287`, `Logo 5`, `Logo 211`, and `Logo 257`). Here, g refers to grid size. (Color figure online)



(a) $g = 3$ (b) $g = 10$ (c) $g = 20$

(d) $g = 6$ (e) $g = 12$ (f) $g = 20$

(g) $g = 3$ (h) $g = 10$ (i) $g = 20$

Fig. 6. $OH(A)$ and $OCS(A)$ on another set of objects (`Logo 23`, `Logo 220`, and `Logo 347`). Here, g refers to grid size. (Color figure online)

From these three measures, it can be said that either A_{in} or A_{out} is convex and concavities in the boundary are very less. If `Logo 5` is considered for the three grid sizes, it is seen that the value ζ_3 increases with the increase of grid size but ζ_1 and ζ_2 decreases for the same. It implies that the difference in the boundary complexity for A_{in} and A_{out} is very less for the lower grid sizes which detects almost all the concavities. For higher grid sizes, the difference increases as some of the concavities are not being detected in both. The nature of ζ_4 is same with ζ_1. The value of ζ_5 remains high for the three grid sizes in `Logo 347`. It means that there is less difference in shape while considering orthogonal convex hull and skull. If the range of ζ_5 is high for lower and higher grid sizes then there is a huge difference in shape for orthogonal convex hull and skull (consider `Logo 181`).

Table 1. Area and perimeter of digital object, A_{out}, A_{in}, $OH(A)$, and $OCS(A)$ (areas are represented as a, a_o, a_i, a_h, and a_s respectively, whereas perimeters are represented as p, p_o, p_i, p_h, and p_s respectively) for the mentioned grid size shown in Figs. 4, 5 and 6. ζ_1, ζ_2, and ζ_3 are boundary complexity measures. Shape complexity measures are ζ_4 and ζ_5.

	g	a_o	a_i	a_h	a_s	p_o	p_i	p_h	p_s	ζ_1	ζ_2	ζ_3	ζ_4	ζ_5
Logo 844	3	37189	32662	45700	19531	1512	1494	948	882	78.4	0.93	0.28	63.39	0.43
$a = 34141$	12	43969	25911	50029	15279	1536	1416	984	888	72	0.84	0.21	48.89	0.31
$p = 1500$	18	53317	21313	53317	13663	1008	1152	1008	756	26.4	0.39	1.00	22.41	0.26
Logo 1287	3	24826	21169	31273	15310	1158	1206	840	810	58.24	1.04	0.38	54.93	0.49
$a = 22404$	10	29251	17741	34131	13591	1100	1080	860	780	44.05	0.81	0.38	40.31	0.4
$p = 1226$	20	36581	13681	39261	10761	1160	960	920	720	39.15	0.63	0.30	25	0.27
Logo 5	3	40603	37333	44299	33820	1176	1152	1008	984	29.02	0.73	0.08	18.9	0.76
$a = 38148$	10	44371	33051	46811	29871	1140	1100	1020	940	24.18	0.53	0.11	14.73	0.64
$p = 1158$	20	49781	27681	52121	26041	1160	960	1040	880	17.27	0.34	0.30	10.43	0.50
Logo 23	3	32638	28810	37882	17671	1284	1260	972	906	52.03	0.84	0.21	54.54	0.47
$a = 30040$	10	37631	24811	41401	14731	1260	1220	1000	860	48.44	0.79	0.32	46.11	0.36
$p = 1280$	20	45021	19361	48521	10741	1240	1120	1040	680	50	0.69	0.40	40.35	0.22
Logo 181	3	20302	17146	27421	15985	1056	1008	912	828	31.21	0.62	0.32	45.7	0.58
$a = 18120$	10	24131	13711	30171	12981	1060	820	940	760	17.34	0.46	0.61	37.36	0.43
$p = 1038$	20	28541	9541	34081	9541	1080	680	960	680	11.56	0.31	1.00	30.57	0.28
Logo 211	3	34648	30466	39232	21442	1236	1386	936	888	56.28	0.91	0.22	42.75	0.55
$a = 31832$	10	38161	25041	41281	18121	1120	1280	960	840	42.31	0.74	0.43	31.54	0.44
$p = 1418$	20	44981	20581	46921	15201	1160	1160	1040	800	33.85	0.60	0.60	23	0.32
Logo 220	6	37759	28099	46075	16699	1572	1004	996	924	40.95	1.13	0.36	60.77	0.36
$a = 32442$	12	42517	24793	49177	15109	1512	1200	1008	840	53.93	0.84	0.17	50.38	0.31
$p = 1602$	20	53361	18482	54121	18482	1120	960	1040	960	4.99	0.09	1.00	2.34	0.34
Logo 257	3	25594	22249	27241	18532	1110	1098	912	882	37.03	0.69	0.14	22.9	0.68
$a = 23425$	10	30271	18631	31381	15121	1140	1060	960	840	35.78	0.61	0.44	19.72	0.48
$p = 1118$	20	36541	13662	36921	12822	1080	920	1040	840	10.73	0.18	0.62	5.21	0.35
Logo 347	3	42421	38704	44503	33382	1230	1212	966	936	43.62	0.78	0.27	18.54	0.75
$a = 39946$	10	45851	33951	47391	30041	1100	1100	980	880	27.46	0.53	0.34	13.64	0.63
$p = 1238$	20	52121	28841	51741	27201	1080	880	1040	800	9.69	0.19	0.64	3.15	0.53
Logo 426	3	26821	22897	39835	19387	1314	1260	972	918	52.78	0.84	0.32	68.37	0.49
$a = 24168$	10	31540	18601	43101	16151	1259	1199	1000	900	43.06	0.68	0.21	57.97	0.37
$p = 1296$	15	35926	16967	44836	16007	1200	1079	1020	960	23.07	0.45	0.55	40.84	0.36

6 Conclusions

A simple and novel approach to analyse the complexity of a digital object is presented here. It gives a measure of the complexity of the global shape of the object. The concept of convexity tree to determine shape complexity is completely a novel technique. Various experimental results are given to show the effectiveness of the proposed scheme. As a future direction, some more features of the object can be derived and metrics can be formulated to classify digital objects based on shapes.

References

1. Biswas, A., Dutt, M., Bhowmick, P., Bhattacharya, B.B.: On finding the orthogonal convex skull of a digital object. In: Proceedings of 13th International Workshop on Combinatorial Image Analysis: IWCIA 2009, pp. 25–36. Progress in Combinatorial Image Analysis, Research Publishing Services, Playa del Carmen, Mexico, November 2009
2. Biswas, A., Sarkar, A., Bhowmick, P., Bhattacharya, B.B.: Combinatorial construction of the orthogonal concavity tree of a digital object. In: 2011 2nd International Conference on Emerging Applications of Information Technology (EAIT), pp. 210–213 (2011)
3. Biswas, A., Bhowmick, P., Bhattacharya, B.B.: TIPS: on finding a tight isothetic polygonal shape covering a 2D object. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 930–939. Springer, Heidelberg (2005). doi:10.1007/11499145_94
4. Biswas, A., Bhowmick, P., Bhattacharya, B.B.: Construction of isothetic covers of a digital object: a combinatorial approach. J. Vis. Commun. Image Represent. **21**(4), 295–310 (2010)
5. Biswas, A., Bhowmick, P., Sarkar, M., Bhattacharya, B.B.: A linear-time combinatorial algorithm to find the orthogonal hull of an object on the digital plane. Inf. Sci. **216**, 176–195 (2012)
6. Biswas, A., Bhowmick, P., Sarkar, M., Bhattacharya, B.B.: Finding the orthogonal hull of a digital object: a combinatorial approach. In: Brimkov, V.E., Barneva, R.P., Hauptman, H.A. (eds.) IWCIA 2008. LNCS, vol. 4958, pp. 124–135. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78275-9_11
7. Dutt, M., Biswas, A., Bhowmick, P., Bhattacharya, B.B.: On finding an orthogonal convex skull of a digital object. Int. J. Imaging Syst. Technol. **21**(1), 14–27 (2011)
8. Hung, K., Aw-Yong, M.: A content-based image retrieval system integrating color, shape and spatial analysis. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, pp. 1484–1488 (2000)
9. Hussein, E., Nakamura, Y., Ohta, Y.: Analysis of detailed patterns of contour shapes using wavelet local extrema. In: Proceedings of 13th International Conference on Pattern Recognition: ICPR 1996, vol. 2, pp. 335–339 (1996)
10. Khan, J.F., Bhuiyan, S.M.A., Adhami, R.R.: Image segmentation and shape analysis for road-sign detection. IEEE Trans. Intell. Transp. Syst. **12**(1), 83–96 (2011)
11. Liu, W., Srivastava, A., Klassen, E.: Joint shape and texture analysis of objects boundaries in images using a Riemannian approach. In: Proceedings of International Conference on Digital Image Processing, vol. 14, pp. 833–837 (2008)
12. Naser, M.A., Hasnat, M., Latif, T., Nizamuddin, S., Islam, T.: Analysis and representation of character images for extracting shape based features towards building an OCR for Bangla script. In: Proceedings of International Conference on Digital Image Processing, pp. 330–334 (2009)
13. Rosenfeld, A., Kak, A.C. (eds.): Digital Picture Processing, 2nd edn. Academic Press, Cambridge (1982)
14. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 1349–1380 (2001)
15. Ulupanar, F., Nevatia, R.: Using symmetries for analysis of shape from contour. In: Proceedings of 2nd International Conference of Computer Vision: ICCV 1998, pp. 414–426 (1998)