

Construction of an Approximate 3D Orthogonal Convex Skull

Nilanjana Karmakar^(✉) and Arindam Biswas

Department of Information Technology,
Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India
nilanjana.nk@gmail.com, barindam@gmail.com

Abstract. Orthogonal convex skull of a 3D digital object is a maximal volume orthogonal convex polyhedron lying entirely inside the object. An efficient combinatorial algorithm to construct an approximate 3D orthogonal convex skull of a digital object is presented in this paper. The 3D orthogonal inner cover, an orthogonal polyhedron which tightly inscribes the digital object, is divided into slab polygons and 2D orthogonal skulls of these slab polygons are combined together using combinatorial techniques to obtain an approximate 3D orthogonal convex skull. The algorithm operates in integer domain and requires at most two passes. The current version of the algorithm deals with non-intersecting objects free from holes and cavities. Experimentation on a wide range of digital objects has provided expected results, some of which are presented here to demonstrate the efficacy of the algorithm.

Keywords: Approximate 3D orthogonal convex skull · Orthogonal slicing · 3D orthogonal inner cover · 3D concavity

1 Introduction

Shape description of digital objects is a prominent area of research in the realm of image analysis. Convex skull can be used as an effective tool for shape description of digital objects. The concept of convex skull was initially studied as the potato-peeling problem which dealt with finding the convex polygon of maximum area contained in a given simple (non-convex) polygon [7, 9]. The solution for a planar n -gon, $n \leq 5$ [9] was followed by polynomial time algorithms of $O(n^9 \log n)$ and $O(n^7)$ [5, 6]. The same problem has been addressed later under the name of the convex skull problem [12]. Variations of the potato-peeling problem using triangulated polygons with or without holes is addressed in [2] and a near-optimal near-linear time algorithm based on visibility graph is proposed in [4]. An orthogonal version of the problem is addressed in [13] with an improved complexity of $O(n^2)$, where the maximal-area orthoconvex polygon is determined by computing the maximal ‘staircase’ boundary of the polygon. The method of finding the orthogonal convex skull of a digital object used in the current work has been reported in [8]. The convex skull problem has been extended to the

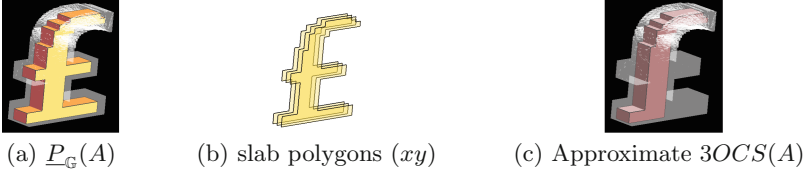


Fig. 1. 3D orthogonal inner cover of a digital object A , slab polygons due to orthogonal slicing along the xy -plane, and an approximate 3D orthogonal convex skull.

3D orthogonal domain where the maximal volume convex subset enclosed in the object is determined by using a constrained distance transform [3].

A novel and efficient algorithm for the construction of an approximate 3D orthogonal convex skull of a digital object is presented in this paper. A two-pass algorithm is used to determine the approximate orthogonal convex skull irrespective of the object size or grid resolution. The algorithm accounts for non-intersecting objects free from holes and cavities. Figure 1(a) shows a digital object, its slice polygons for xy -plane (Fig. 1(b)), and its approximate $3OCS(A)$ is shown in Fig. 1(c). The rest of the paper is organized as follows. The problem is defined in Sect. 2. The algorithm with its time complexity are explained in Sect. 3. The paper is concluded with experimental results in Sect. 4.

2 Definitions and Preliminaries

A *digital object* A is defined as a finite subset of \mathbb{Z}^3 , with all its constituent points (i.e., voxels) having integer coordinates and connected in 26-neighborhood. Each voxel is equivalent to a *3-cell* [11] centered at the concerned integer point (Fig. 2(Left)). A *digital grid* \mathbb{G} consists of three orthogonal sets of equi-spaced grid lines along the x -, y -, and z -axes. A larger (smaller) value of the grid size g implies a sparser (denser) grid. A *unit grid cube* (UGC) is a (closed) cube of length g . A UGC-face, f_k , has two adjacent UGCs, U_1 and U_2 , such that $f_k = U_1 \cap U_2$ (Fig. 2(Left)). A UGC consists of $g \times g \times g$ voxels and each UGC-face consists of $g \times g$ voxels.

An *orthogonal polyhedron* is a 3D polytope with all its vertices as grid vertices, all its edges made of grid edges, and all its faces being simple isothetic polygons lying on face planes. An *orthogonal convex polyhedron* is an orthogonal polyhedron whose intersection with a face plane parallel to any coordinate plane is either empty or a collection of projection-disjoint orthogonal convex polygons¹. The *3D orthogonal inner cover* of A , $\underline{P}_G(A)$, is defined as the set of orthogonal polyhedrons that tightly inscribes A ; i.e.,

- i. $\underline{P}_G(A) \subseteq A$
- ii. for each $p \in \underline{P}_G(A)$, $0 < d_T(p, A') \leq g$

¹ Orthogonal convex polygons are also known as “*hv-convex*” polygon in literature..

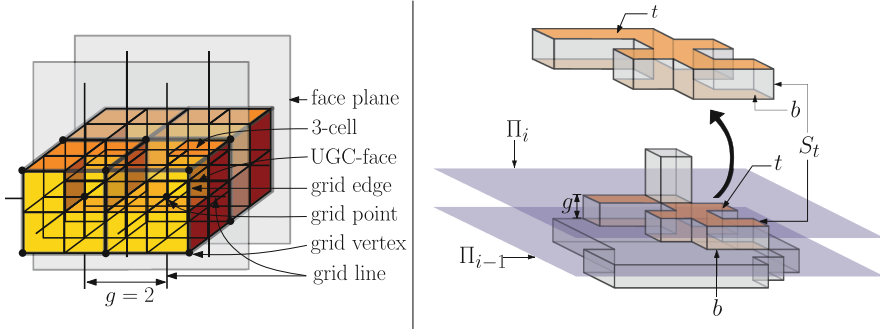


Fig. 2. Left: α -adjacent 3-cells for $g = 2$. **Right:** Slab S_t of height g , bounded on top by slab polygon t lying on Π_i and in the bottom by b , the projection of t on Π_{i-1} .

where $\underline{P}_G(A)$ denotes the entire inner cover including its surface $\underline{P}_G(A)$ and interior region. Here, $A' = \mathbb{Z}^3 \setminus A$ and d_\top denotes isothetic distance². In this work, we consider objects such that its $\underline{P}_G(A)$ contains only one orthogonal polyhedron.

The *3D orthogonal convex skull* of a digital object A , denoted by $3OCS(A)$, is a maximal volume orthogonal polyhedron such that

- i. no point $p \in \mathbb{Z}^3 \setminus A$ lies on or inside $3OCS(A)$ and
- ii. $3OCS(A)$ is orthogonally convex.

3 Proposed Work

Given a digital object, A , its inner orthogonal cover, $\underline{P}_G(A)$, is sliced into slab polygons [10] (Sect. 3.1) along one plane (say xy -). The concavities in these slab polygons are removed and the convex slab polygons are regrouped to form an orthogonal polyhedron. The resulting polyhedron is again sliced along another plane (say yz -) and the concavities are removed from the slab polygons. Similarly, this procedure is repeated for zx -plane. This constitutes one pass of the algorithm. After the second pass, the resulting orthogonal polyhedron, which is devoid of any concavity along any plane, is an approximate orthogonal convex skull.

3.1 Slicing and Orthogonal Slabs

The 3D object A is provided as a set of voxels. A is imposed on a 3D digital grid \mathbb{G} represented as a set of UGCs each of grid size g . A UGC U_l containing object

² *Isothetic distance* between two points $p(x_1, y_1, z_1)$ and $q(x_2, y_2, z_2)$ is defined as the Minkowski norm L_∞ given by $d_\top(p, q) = \max\{|x_1 - x_2|, |y_1 - y_2|, |z_1 - z_2|\}$. Isothetic distance of a point p from an object A is $d_\top(p, A) = \min\{d_\top(p, q) : q \in A\}$. It may be noted that isothetic distance may also be expressed in terms of Chebyshev distance [1] which is a special case of Minkowski norm..

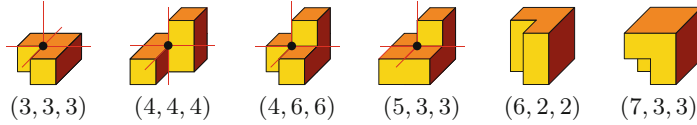


Fig. 3. Types of concave vertex (that do not belong to intersecting polyhedron).

voxels is defined as a *partially object-occupied* UGC. A UGC U_k lying completely inside A is defined as a *fully object-occupied* UGC.

Let $\Pi = \{\Pi_1, \Pi_2, \dots, \Pi_r\}$ be a set of *slicing planes* separated by g and parallel to the zx -plane (or yz - or xy -plane) which intersects $\underline{P}_G(A)$. A UGC-face f_k (f_l) is considered as *fully object-occupied* (*partially object-occupied*) if there exists a U_k (U_l) below (in case of zx -plane) f_k (f_l). The inner boundary of A intersected by Π_i is traversed orthogonally keeping fully occupied UGC-faces f_k to the left. Thus, a *slab polygon* on Π_i is obtained. Let t be a slab polygon on Π_i and b be the projection of t on Π_{i-1} . The section of $\underline{P}_G(A)$ of height g intercepted between Π_i and Π_{i-1} and bounded horizontally on top and bottom by t and b respectively is defined as the slab S_t (Fig. 2(Right)). Since b is the projection of t , their shapes are identical, that is, t does not vary from Π_i to Π_{i-1} . Hence, S_t can be represented by t . It is evident that the UGCs contained in a slab are fully object-occupied.

3.2 Concavity in Three Dimensions

Depending on the fully object-occupied neighboring UGCs a grid vertex v may be classified into different types where each type is represented by a 3-tuple defined as (#incident UGCs, #incident edges, #incident faces) w.r.t. v . The grid vertices of types (3, 3, 3), (4, 4, 4), (4, 6, 6), (5, 3, 3), (6, 2, 2), and (7, 3, 3) are classified as concave vertices. In Fig. 3, some instances of the possible concave vertices (which do not form intersecting polyhedrons) are shown.

While traversing a slab polygon t (which represents S_t) a concavity is detected if we encounter at least two consecutive concave vertices. In Fig. 4(a), the concave vertices on t and b are shown in blue color. For a nested concavity the number of consecutive concave vertices is more than two, as shown in Fig. 4(c). The rectangular faces of S_t having width g and incident on the concave vertices are defined as *concavity faces*. If two concavity faces are parallel to each other, then they are referred to as parallel concavity faces. A concavity on a slab has at least three concavity faces. Two of them must be parallel concavity faces (see Fig. 4(a) and (c)).

3.3 Resolving the Concavities of a Slab

During the traversal along the boundary of a slab, whenever a concavity is detected, it is resolved as follows. A face plane Π_f perpendicular to a slab S_t and passing through the concave vertices of a concavity divides the slab into

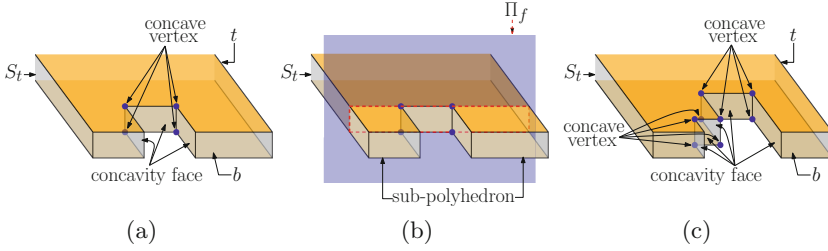


Fig. 4. (a) A concavity on slab S_t containing two consecutive concave vertices (blue) on each of the slab polygons t and b , and a pair of parallel concavity faces. (b) Sub-polyhedrons formed when a face plane Π_f passes through the concave vertices of S_t . (c) Nested concavity on S_t containing four consecutive concave vertices on each of t and b (Color figure online).

three different parts: two separate sub-polyhedrons lying on one side of Π_f , and the rest of the slab on the other side (Fig. 4(b)). To maintain convexity of the slab one of the sub-polyhedrons has to be dropped depending on whether the concavity is defined by two or more consecutive 270° grid vertices. While traversing a concavity, the sub-polyhedron appearing before the concavity has already been processed in the previous steps. Hence, that sub-polyhedron does not contain any concavity. The next sub-polyhedron is checked recursively for concavity. If deleting a sub-polyhedron disconnects the slab into two parts, then the sub-polyhedron is not deleted. Otherwise, the sub-polyhedron having the smaller volume is dropped. This ensures that the sub-polyhedron having the larger volume is included in $OCS(S_t)$, thereby striving to achieve a larger volume of $3OCS(A)$. As retaining the larger volume sub-polyhedron is a local decision, it does not ensure that $3OCS(A)$ will be of the largest possible volume.

Figure 5 shows a brief demonstration of resolving the concavities in a slab. Concavity C_1 (category $\mathbf{C}_{z,x}$) has two sub-polyhedrons \mathcal{A} and \mathcal{B} (Fig. 5(a)). The sub-polyhedron \mathcal{A} , occurring after C_1 , is checked recursively for concavity. Concavity C_2 (category $\mathbf{C}_{z,y}$) is detected on \mathcal{A} (Fig. 5(b)). C_2 has sub-polyhedrons \mathcal{A}_1 and \mathcal{A}_2 . \mathcal{A}_2 is devoid of concavity. Volume of \mathcal{A}_2 is smaller than \mathcal{A}_1 and deleting \mathcal{A}_2 does not disconnect the slab. Hence, \mathcal{A}_2 is deleted. In case of concavity C_3 (category $\mathbf{C}_{z,x}$), sub-polyhedrons \mathcal{A}_3 and \mathcal{A}_4 are of equal volume (Fig. 5(c)). As deleting \mathcal{A}_4 disconnects the slab, \mathcal{A}_4 is not deleted. Hence, \mathcal{A}_3 is deleted. Now, sub-polyhedron \mathcal{A} of concavity C_1 contains no further concavity (Fig. 5(d)). As volume of \mathcal{B} is less than \mathcal{A} , sub-polyhedron \mathcal{B} is deleted to resolve C_1 (Fig. 5(e)). Similarly, concavity C_4 (category $\mathbf{C}_{z,y}$) is resolved by deleting sub-polyhedron \mathcal{D} . Thus, all the concavities on the given slab are resolved (Fig. 5(f)). The process is repeated for all the slabs parallel to a given coordinate plane. After each deletion of a sub-polyhedron, $\underline{P}_G(A)$ is modified accordingly.

3.4 Finding Approximate 3D Orthogonal Convex Skull

Construction of an approximate 3D orthogonal convex skull of A involves resolving the concavities of $\underline{P}_G(A)$ so that $3OCS(A)$ is a convex orthogonal polyhedron. Along each coordinate plane the following steps are carried out:

- i. A is sliced orthogonally to form a set of orthogonal slabs that represent $\underline{P}_G(A)$ (Sect. 3.1).
- ii. The concavities on each slab are detected and resolved, thereby modifying $\underline{P}_G(A)$ (Sect. 3.3).
- iii. If $\underline{P}_G(A)$ has been disconnected into more than one polyhedrons, then all the polyhedrons except the one having the largest volume are discarded.

The above steps are repeated along another coordinate plane considering the modified $\underline{P}_G(A)$ as input.

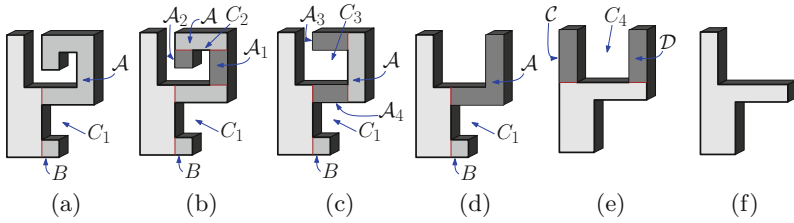
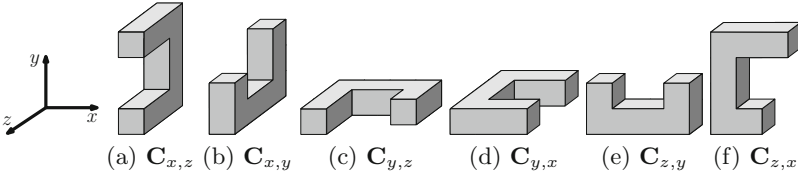


Fig. 5. Demonstration of resolving the concavities on a slab. Concavity C_1 is of category $C_{z,x}$, C_2 of category $C_{z,y}$, C_3 of category $C_{z,x}$, and C_4 of category $C_{z,y}$.



Plane	Concavity	Induced concavity		
		yz -plane	zx -plane	xy -plane
yz	$C_{x,z}$	$C_{y,z}$	-	-
	$C_{x,y}$	$C_{z,y}$	-	-
zx	$C_{y,z}$	-	$C_{x,z}$	-
	$C_{y,x}$	-	$C_{z,x}$	-
xy	$C_{z,y}$	-	-	$C_{x,y}$
	$C_{z,x}$	-	-	$C_{y,x}$

Fig. 6. Two categories of possible concavities w.r.t. a slab along each coordinate plane. (a) Resolving concavity of category $C_{x,z}$ induces concavity of category $C_{y,z}$ along yz -plane, (b) resolving $C_{x,y}$ induces $C_{z,y}$ along yz -plane, (c) resolving $C_{y,z}$ induces $C_{x,z}$ along zx -plane, (d) resolving $C_{y,x}$ induces $C_{z,x}$ along zx -plane, (e) resolving $C_{z,y}$ induces $C_{x,y}$ along xy -plane, and (f) resolving $C_{z,x}$ induces $C_{y,x}$ along xy -plane.

Along each coordinate plane there exists exactly two categories of concavity on a slab S_t , as shown in the table in Fig. 6. A concavity can be resolved only along the coordinate plane to which it is parallel, i.e., $C_{x,z}$ (belonging to category $\mathbf{C}_{x,z}$) and $C_{x,y}$ (belonging to category $\mathbf{C}_{x,y}$) are resolved only along the yz -plane, etc. Resolving a concavity along a coordinate plane may induce another concavity along a different coordinate plane which leads to the following observation.

Observation 1. *W.l.o.g., resolving an instance of concavity of category $\mathbf{C}_{x,z}$, while traversing the slab along the yz -plane, may induce one (or more) instance(s) of concavity of category $\mathbf{C}_{y,z}$.*

It is observed that resolving a concavity of category $\mathbf{C}_{x,z}$ may induce a concavity of category $\mathbf{C}_{y,z}$, and resolving a concavity of category $\mathbf{C}_{y,z}$ may induce a concavity of category $\mathbf{C}_{x,z}$; resolving $\mathbf{C}_{x,y}$ may induce $\mathbf{C}_{z,y}$ and vice versa; resolving $\mathbf{C}_{y,x}$ may induce $\mathbf{C}_{z,x}$ and vice versa. W.l.o.g. let us consider concavity $C_{y,z}$ (belongs to $\mathbf{C}_{y,z}$). $C_{y,z}$ and $C_{x,z}$ (belongs to $\mathbf{C}_{x,z}$) may be induced from each other for a finite number of times, which leads to the following lemma.

Lemma 1. *Let resolving an instance of concavity $C_{y,z}$ induces one or more instances of concavity $C_{x,z}$ and resolving those instances of $C_{x,z}$ induces one or more instances of $C_{y,z}$. Then resolving the instances of the induced concavity $C_{y,z}$ does not induce any further concavity.*

Proof. Let the concavities of slab polygons of P along the zx -plane be resolved first, followed by the yz -plane and the xy -plane. Since resolving a concavity may induce another concavity, more than one pass may be required to resolve the induced concavities, as will be elaborated later in Theorem 1. In the first pass, let the concavity $C_{y,z}$ on slab Sy_1 of P (Fig. 7(a)) be resolved by deleting one of its sub-polyhedrons \mathcal{A} along the zx -plane (Fig. 7(b)). As a result one or more instances of concavity $C'_{x,z}$ may be induced on slabs Sx_1 and Sx_2 of P' , by Observation 1 (Fig. 7(c)). While resolving each instance of $C'_{x,z}$ along the yz -plane (Fig. 7(d)), one or more instances of concavity $C''_{y,z}$ may be induced on slabs Sy_2 and Sy_3 of P'' (Fig. 7(e)). In this case, no concavity is detected along the xy -plane in the first pass. Hence, it is not shown in Fig. 7. In the second pass, one or more instances of concavity $C'_{y,z}$ on P'' (Fig. 7(f)) are resolved along the zx -plane. Let us assume, by contradiction, that this induces one or more instances of $C'''_{x,z}$ on slab Sx_3 of P''' (Fig. 7(h)).

A concavity on a slab is characterized by two sub-polyhedrons and is resolved by deleting any one of them according to certain rules (Sect. 3.3). During the first pass along the zx -plane, $C_{y,z}$ is the only concavity detected on slab Sy_1 of P (Fig. 7(a)). During the second pass along the zx -plane (Fig. 7(e)), two instances of concavity $C'_{y,z}$ are detected on slabs Sy_2 and Sy_3 . They are resolved by deleting sub-polyhedrons \mathcal{D} and \mathcal{E} respectively from P'' . Since $C'_{y,z}$ is not detected on P in the first pass along the zx -plane, it is concluded that the sub-polyhedrons \mathcal{D} and \mathcal{E} existed as a part of P (Fig. 7(a)). Hence, it is justified that if resolving an instance of concavity $C_{y,z}$ induces one or more instances of concavity $C_{x,z}$, then resolving those instances of $C_{x,z}$ induces one or more instances of $C_{y,z}$ (Fig. 7 (a-e)).

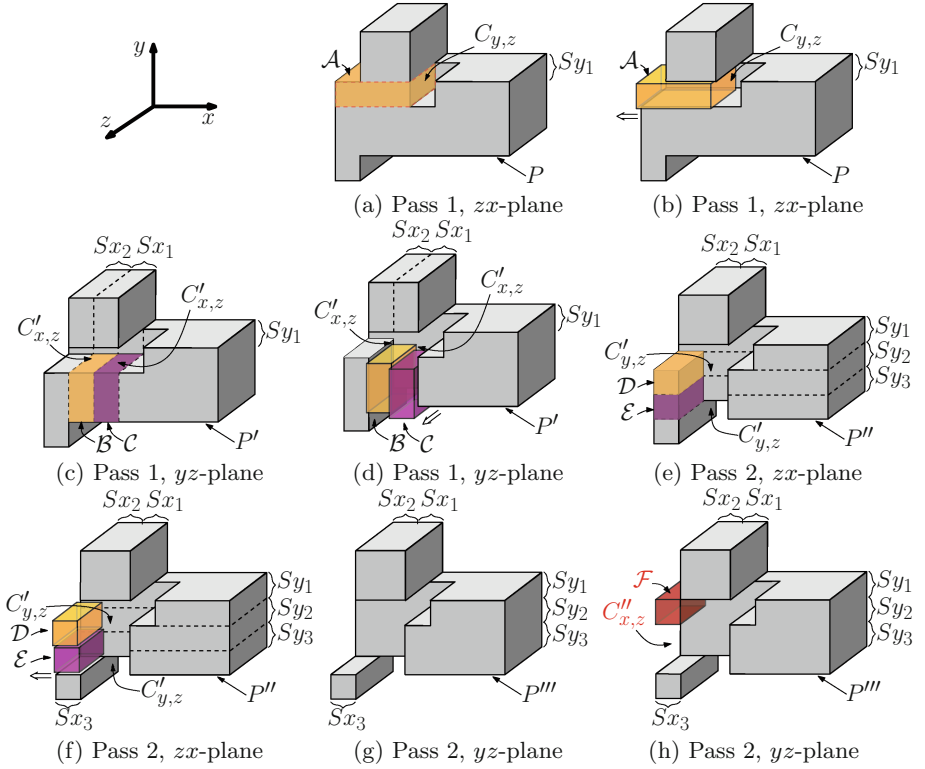


Fig. 7. (a) When an instance of concavity $C_{y,z}$ is resolved along the zx -plane (b) by deleting a sub-polyhedron A , (c) two instances of concavity $C'_{x,z}$ are induced, which are resolved along the yz -plane (d) by deleting sub-polyhedrons B and C . As a result (e) two instances of $C'_{y,z}$ are induced which are resolved along the zx -plane (f) by deleting sub-polyhedrons D and E . (g) Polyhedron P''' is obtained after resolving all the concavities of P . (h) It is assumed that an instance of concavity $C''_{x,z}$ exists on P''' .

During the first pass along the yz -plane (Fig. 7(c)), two instances of concavity $C'_{x,z}$ are detected on the slabs Sx_1 and Sx_2 of P' . If it is assumed that concavity $C''_{x,z}$ exists on slab Sx_3 (Fig. 7(h)), then the sub-polyhedron \mathcal{F} should be present on the slab Sx_3 of P''' during the second pass along the yz -plane. But the sub-polyhedron \mathcal{F} did not exist as a part of P' during the first pass along the yz -plane (Fig. 7(c)). It implies that \mathcal{F} has been deleted before the first pass along the yz -plane, i.e., \mathcal{F} has been deleted while resolving $C_{y,z}$ along the zx -plane (Fig. 7(b)). Hence, \mathcal{F} cannot exist on Sx_3 during the second pass along the yz -plane. In absence of the sub-polyhedron \mathcal{F} , concavity $C''_{x,z}$ cannot exist on Sx_3 of P''' (Fig. 7(h)). This contradicts our assumption. Hence, no concavity is induced while resolving an instance of $C'_{y,z}$ (Fig. 7(g)). Hence proved. \square

The induced concavities may not be resolved within a single pass of the algorithm, i.e., applying the algorithm once along the yz -, zx -, and xy -planes. The following theorem proves that two passes of the algorithm are sufficient for the purpose.

Theorem 1. *The orthogonal polyhedron formed by applying the proposed algorithm at most twice on $\underline{P}_G(A)$ for the set of three coordinate planes (yz -, zx -, and xy -planes) gives an approximate 3D orthogonal convex skull $3OCS(A)$.*

Proof. According to Lemma 1, a concavity along with its subsequent induced concavities (if any) are resolved completely by applying the concavity removal method along three coordinate planes. The sequence of coordinate planes starts and ends with the same coordinate plane. For example, a concavity of category $C_{y,z}$ and the induced concavities of category $C_{x,z}$ and again $C_{y,z}$ are resolved along the zx -, yz -, and zx -planes (Fig. 7). This is possible only if two passes of the algorithm are used. Since there exists categories of concavity along each of the three coordinate planes, application of the algorithm along each of yz -, zx -, and xy -planes may be required twice. Resolving all the concavities of $\underline{P}_G(A)$ can, however, conclude in less than two passes depending on the object structure which does not induce concavity. Therefore, all the concavities and induced concavities on $\underline{P}_G(A)$ are resolved in at most two passes of the algorithm.

A slab refers to a section of $\underline{P}_G(A)$ at a given slicing plane. Since $\underline{P}_G(A)$ is of maximum volume that can be inscribed in the object, at the given slicing plane a slab is also of maximum volume. While resolving a concavity on a slab, the sub-polyhedron with the larger volume is included in the 3D orthogonal convex skull of the slab, thereby trying to achieve a larger volume of $3OCS(A)$. It may be noted that the 3D orthogonal convex skull of the slab may not be unique due to the varying starting point and initial direction of traversal. Also, the volume of $3OCS(A)$ may vary with different order of the coordinate planes along which the algorithm is applied. Due to the variation in volume, the result given by the algorithm is an approximate 3D orthogonal convex skull. Hence proved. \square

3.5 Algorithm

Given an object A as a set of voxels, its approximate 3D orthogonal convex skull is constructed by the two-pass algorithm presented in Fig. 8. The three coordinate planes are considered in sequence (**for** loop Steps 3–15), for each slicing plane along a coordinate plane the concavities in a slab are removed (Steps 6–11). On a slicing plane, each slab $S[i]$ is subjected to a method explained in Sect. 3.3 to construct the 2D orthogonal convex skull $OCS(S[i])$ (Step 10). Consequently, the slab corresponding to the 2D orthogonal convex skull $K[i]$ along each slicing plane is accumulated in $\underline{P}'_G(A)$ (Step 11).

While finding the 2D orthogonal convex skull of the slab polygon w.r.t. a slab, the volume of the slab is determined by computing the area of the slab polygon. The total volume of a set of consecutive slabs in the direction perpendicular to

```

Algorithm 3OCS( $A, \mathbb{G}$ )
01.  $count \leftarrow 0$   $\triangleright$  # passes
02. do
03.   for each coordinate plane  $j$   $\triangleright j \in \{yz, zx, xy\}$ 
04.     set  $\underline{P}'_{\mathbb{G}}(A) \leftarrow 0$   $\triangleright$  empty set of polyhedrons
05.     slice  $\underline{P}'_{\mathbb{G}}(A)$  with  $\Pi$   $\triangleright \Pi = \{\Pi_1, \Pi_2, \dots, \Pi_r\}$ 
06.     for each slicing plane  $\Pi_k$   $\triangleright 1 \leq k \leq r$ 
07.       set  $s \leftarrow$  # slabs on  $\Pi_k$ 
08.       set  $S \leftarrow$  set of slabs on  $\Pi_k$ 
09.       for  $i = 1$  to  $s$ 
10.         set  $K[i] \leftarrow$  2D orthogonal convex skull  $OCS(S[i])$ 
11.          $\underline{P}'_{\mathbb{G}}(A) \leftarrow \underline{P}'_{\mathbb{G}}(A) \cup K[i]$ 
12.       set  $m \leftarrow$  # polyhedrons in  $\underline{P}'_{\mathbb{G}}(A)$ 
13.       set  $M \leftarrow$  set of polyhedrons in  $\underline{P}'_{\mathbb{G}}(A)$ 
14.        $\underline{P}_{\mathbb{G}}(A) \leftarrow largest(vol(M[1]), vol(M[2]), \dots, vol(M[m]))$ 
15.    $count \leftarrow count + 1$ 
16. while( $count < 2$ )
17. return  $\underline{P}_{\mathbb{G}}(A)$ 

```

Fig. 8. Brief outline of the proposed algorithm.

the given coordinate plane gives the volume of the polyhedron composed of the set of slabs. Thus, the volumes of all the orthogonal polyhedrons in $\underline{P}'_{\mathbb{G}}(A)$ are determined. If $\underline{P}'_{\mathbb{G}}(A)$ contains a single polyhedron, then $\underline{P}'_{\mathbb{G}}(A)$ represents the modified 3D orthogonal inner cover $\underline{P}_{\mathbb{G}}(A)$. If $\underline{P}'_{\mathbb{G}}(A)$ contains more than one polyhedron, then the polyhedron having the largest volume is assigned to $\underline{P}_{\mathbb{G}}(A)$ (Step 14) and the rest of the polyhedrons are discarded.

The process is repeated with the modified $\underline{P}_{\mathbb{G}}(A)$ along the other coordinate planes (Steps 3–15) and the algorithm is repeated exactly once along all the three coordinate planes (Steps 2–16). Finally, the modified 3D orthogonal inner cover $\underline{P}_{\mathbb{G}}(A)$ is reported as the approximate 3D orthogonal convex skull (approximate 3OCS(A)) (Step 17). The algorithm deals with non-intersecting objects free from holes and cavities.

3.6 Time Complexity

Let n be the number of voxels on the object surface connected in 26-neighborhood. A UGC is a cube of length g which contributes a maximum of five faces to the cover. Therefore, the number of UGCs on the object surface containing object voxels is $O(n/g)$ in the worst case, which implies that the number of UGC-faces on the object surface is given by $O(n/g)$. The full object occupancy of a UGC is determined by checking six UGC-faces completely in $O(g^2)$ time.

W.r.t. each slicing plane, orthogonal slicing involves traversal of the grid vertices on the slicing plane exactly once. Therefore, considering all the slicing planes, the UGCs on the object surface are traversed exactly once. This traversal requires $O(n/g)$ time. $O(g^2)$ time is required to check whether a UGC-face is fully object-occupied. Hence, the direction of traversal at each grid vertex is

determined in $O(g^2)$ time. Therefore, the orthogonal slicing along a coordinate plane is completed in $O(n/g) \times O(g^2) = O(n/g)$ time.

The grid vertices on the object surface are sorted lexicographically in $O((n/g) \log(n/g))$ time. Once a concavity is detected on a slab, the terminal vertices of the next sub-polyhedron are found from the lexicographically sorted lists. For all the slabs this operation is completed in $O(\log n)$ time. In case of nested concavity, connectivity of a sub-polyhedron is checked in $O(1)$ time. Volumes of the sub-polyhedrons are computed in $O(n/g)$ time to remove the sub-polyhedron of smaller volume in every case. Hence, the overall time complexity for resolving the concavities on all the slabs is bounded by $O(n \log n)$. Volume of the approximate 3D orthogonal convex skull is determined by computing the volume of all the slabs parallel to a given coordinate plane in $O(n/g)$ time. Therefore, the total time complexity for finding an approximate $3OCS(A)$ is given by $O(n/g) + O(n \log n) = O(n \log n)$.

4 Experimental Results and Conclusions

The proposed algorithm has been implemented in C in Linux Fedora Release 13, Dual Intel Xeon Processor 2.8 GHz, 800 MHz FSB. The experimental results in



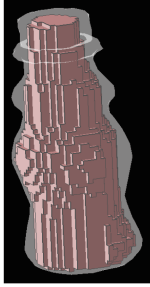
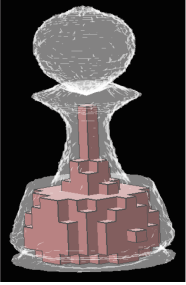
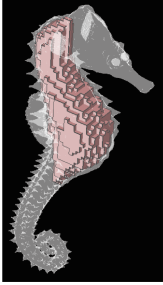

Chess pawn	Seahorse	Bottle
		
$g = 2, vol_{OIC} = 4630,$ $vol_{OCS} = 3628$	$g = 2, vol_{OIC} = 52690,$ $vol_{OCS} = 36583$	$g = 2, vol_{OIC} = 14967,$ $vol_{OCS} = 13605$
		
$g = 4, vol_{OIC} = 436,$ $vol_{OCS} = 365$	$g = 4, vol_{OIC} = 5445,$ $vol_{OCS} = 3964$	$g = 4, vol_{OIC} = 1524,$ $vol_{OCS} = 1374$

Fig. 9. Approximate 3D orthogonal convex skull of Chess pawn, Seahorse, and Bottle.

Fig. 9 display an approximate 3D orthogonal convex skull of each of the objects **Chess pawn**, **Seahorse**, and **Bottle** for different grid sizes. The volumes of both the 3D orthogonal inner cover (vol_{OIC}) and the approximate 3D orthogonal convex skull (vol_{OCS}) decrease exponentially with increasing grid size. It may be noted that the number of concave vertices of all the types except $(7, 3, 3)$ is less in the approximate $3OCS(A)$ than in the 3D orthogonal inner cover.

The approximate 3D orthogonal convex skull of an object may vary with the order of the coordinate planes along which the algorithm is applied. If the two sub-polyhedrons due to a concavity on a slab are of equal volume, then more than one result may exist. The approximate 3D orthogonal convex skull may also vary depending on the starting point and initial direction of traversal (anti-clockwise or clockwise) while resolving the concavities on a slab. The result will be unique only if none of the concavities on a slab have more than two consecutive 270° vertices or the sub-polyhedron with the larger volume is not deleted to maintain connectivity of the slab. The volume of $3OCS(A)$ may vary with its structure, reporting an approximate 3D orthogonal convex skull in every case. Figure 10 illustrates the variation of the approximate 3D orthogonal convex skull when the proposed algorithm is applied on the object along the coordinate planes in different orders, like, along the yz -plane first, followed by the zx -plane and the xy -plane, etc. The current version of the algorithm is limited to non-intersecting digital objects and objects free from holes and cavities. Extension of the algorithm to account for those objects may be attempted in future.

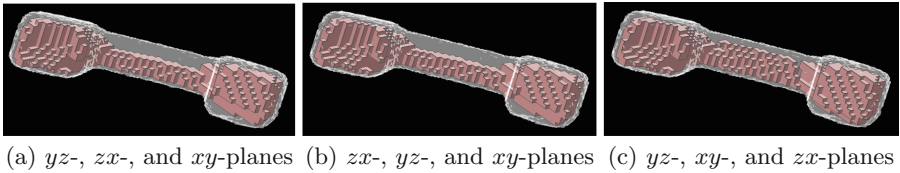


Fig. 10. Variation of the approximate 3D orthogonal convex skull of Phone due to application of the algorithm along the coordinate planes in different orders.

Acknowledgement. A part of this research is funded by CSIR, Govt. of India.

References

1. Abello, J.M., Pardalos, P.M., Resende, M.G.C. (eds.): Handbook of Massive Data Sets, vol. I. Springer, Heidelberg (2002)
2. Aronov, B., van Kreveld, M., Löffler, M., Silveira, R.I.: Peeling meshed potatoes. *Algorithmica* **60**(2), 349–367 (2011)
3. Borgfors, G., Strand, R.: An approximation of the maximal inscribed convex set of a digital object. In: Roli, F., Vitulano, S. (eds.) *ICIAP 2005*. LNCS, vol. 3617, pp. 438–445. Springer, Heidelberg (2005)

4. Cabello, S., Cibulka, J., Kynčl, J., Saumell, M., Valtr, P.: Peeling potatoes near-optimally in near-linear time. In: Proc. 13th Annual Symposium on Computational Geometry, SOCG 2014, Kyoto, Japan. pp. 224–231. ACM, New York (2014)
5. Chang, J., Yap, C.: A polynomial solution for potato-peeling and other polygon inclusion and enclosure problems. In: Proceedings of the 25th Annual Symposium on Foundations of Computer Science, SFCS 1984, Singer Island, Florida, pp. 408–416. IEEE Computer Society, Washington, DC (1984)
6. Chang, J., Yap, C.: A polynomial solution for the potato-peeling problem. *Discrete Comput. Geom.* **1**(2), 155–182 (1986)
7. Chassery, J.M., Coeurjolly, D.: Optimal shape and inclusion. In: Ronse, C., Najman, L., Decencière, E. (eds.) *Mathematical Morphology: 40 Years On. Computational Imaging and Vision*, vol. 30, pp. 229–248. Springer, Netherlands (2005)
8. 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**, 14–27 (2011)
9. Goodman, J.E.: On the largest convex polygon contained in a non-convex n -gon, or how to peel a potato. *Geometriae Dedicata* **11**(1), 99–106 (1981)
10. Karmakar, N., Biswas, A., Bhowmick, P.: Fast slicing of orthogonal covers using DCEL. In: Barneva, R.P., Brimkov, V.E., Aggarwal, J.K. (eds.) *IWCIA 2012. LNCS*, vol. 7655, pp. 16–30. Springer, Heidelberg (2012)
11. Klette, R., Rosenfeld, A.: *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco (2004)
12. Woo, T.: The convex skull problem. Technical report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI (1986)
13. Wood, D., Yap, C.K.: The orthogonal convex skull problem. *Discrete Comput. Geom.* **3**(4), 349–365 (1988)