Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Computers & Graphics 28 (2004) 907–918

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Technical section

A potential-based generalized cylinder representation☆

Jen-Hui Chuang*, Narendra Ahuja[1], Chien-Chou Lin,
Chi-Hao Tsai, Cheng-Hui Chen

Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30010, ROC

Abstract

Generalized cylinder (GC) is a class of parametric shapes that is very flexible and capable of modeling many different types of real-world objects, and have subsequently been the focus of considerable research in the vision community. Most of the related works proposed previously have dealt with the recovery of 3D shape description of objects based on the GC representation from one or more 2D image data. Different from the objective of the previous works, in this paper, we will propose a new approach to obtain a GC-based shape description of 3D objects. The proposed approach of deriving the GC axis is a further extension of the potential-based skeletonization approach presented in (IEEE Trans. Pattern Anal. Mach. Intell. 22(11) (2000) 1241). Simulation results demonstrate that the derived GC representation will yield better approximation of object shape than that based on simpler subclasses of GC since there is, in principle, no restriction on the topology of the GC axis and the shape of the cross-sections.

Keywords: Generalized cylinder; Potential field; 3D Shape representation

1. Introduction

It is well accepted that a good way to represent a complex 3D object is to decompose it into simpler volumetric parts and then describe the parts and the relationships among them [2]–[5]. A popular representation for such volumetric parts is to use generalized cylinder (GC)-based shape representations [2]. A GC is a solid defined by its axis, a cross-section curve, and a scaling function. The solid is obtained by sweeping a planar region, its cross-section, as the region is moved and deformed according to the scaling function along a space curve, its axis. Fig. 1 shows examples of three simple subclasses of GCs whose axes are planar curves while the cross-sections are orthogonal to the local axis direction. For the straight homogeneous generalized cylinder (SHGC) [6,7], their cross-sections have the same shape but may vary in size along a straight axis. For the planar right constant generalized cylinders (PRCGCs), their cross-sections are fixed both in shape and size while the axis can be any planar curve. For the circular planar right generalized cylinders (circular PRGCs), their axes are planar curves while the cross-sections are fixed in shape (circular) but not in size. In short, each of the above three subclasses has (i) a constant axis direction (SHGC), (ii) a constant size of cross-section (PRCGC), or (iii) a fixed shape of cross-sections (PRGC).

GCs were first proposed by Binford [2] as a class of parametric shapes that is very flexible and capable of modeling many different types of real-world objects, and

☆This work was supported by National Science Council, Republic of China, under grant no. NSC90-2213-E-009-151.

*Corresponding author.

E-mail address: jchuang@cis.nctu.edu.tw (J.-H. Chuang).

[1] N. Ahuja is with the Beckman Institute and Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801, USA.

0097-8493/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2004.08.004

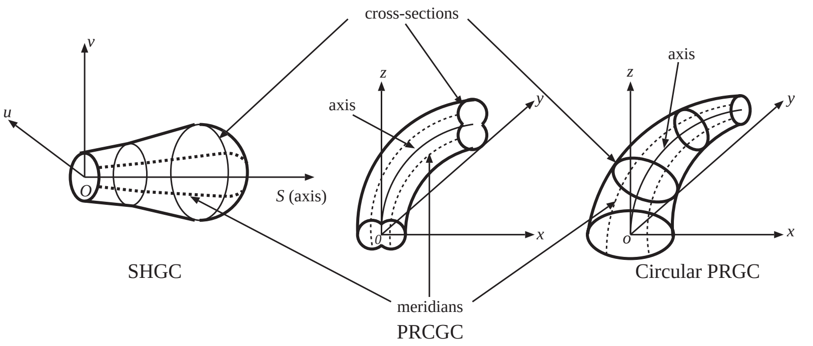


Fig. 1. Three major subclasses of GCs : SHGC, PRCGC, and Circular PRGC.

have subsequently been the focus of considerable research in the vision community. Most of the related works proposed previously have dealt with the recovery of 3D shape description of objects from image data (shape from contour). Recovering 3D shape of objects in a scene from image data remains an important and difficult problem in computer vision. The difficulty arises from the fact that an image is a 2D projection of the scene and the process is not invertible without making some assumptions. Thus, most methods proposed for recovering generalized cylinder descriptions have implicitly or explicitly relied on the assumption that GCs project onto 2D images with some degree of symmetry, and use this assumption as the basis for their recovery algorithms. On the other hand, the algorithms are restricted to some subclasses of GCs such as SHGC, PRCGC, and circular PRGC.

Agin [8], and Nevatia and Binford [3] used occluding (partial viewed) contours to recover generalized cylinder description from range data. Their methods assume implicitly that the occluding contours of a generalized cylinder are symmetric with respect to its axis. Complete generalized cylinder descriptions are obtained by grouping hypothesized local generalized cylinders, whose axes are found as the midpoints of pairs of occluding contour points. Marr proposed an algorithm in [9] to recover shape from GC contours while assuming that the projection is orthographics and that the occluding contours of an observed SHGC (which he called a generalized cone) are planar.

Horaud and Brady [10] proved that the contours of a solid of revolution (SOR), an important subclass of SHGCs, are symmetric with respect to the image of its axis. An algorithm was given for recovering the position and orientation of a solid of revolution from a single image of its contours using this property. They also argued that the method could be extended to the case of an arbitrary SHGC by approximating it by some best-fitting "perceived surface of revolution". Nalwa's method [11] was limited to the analysis of solids of revolution and gave some properties of such objects, but does not provide a method for complete reconstruction. Other recovery methods for surfaces of revolution were also considered by Richetin et al. [12] and LaVest [13].

In [7], Shafer proved several general properties of straight homogeneous generalized cylinders, and derived the general equation that a point must satisfy to belong to the occluding contour of a SHGC, without any restriction on the viewing direction. Rao and Nevatia [14] used Shafer's result that the occluding contours of a linear SHGC are planar and symmetric with respect to its axis to segment sparse range data. Ponce et al. [15] proved several new projective invariant properties of the contours of SHGCs, and used these properties in two implemented algorithms for recovering generalized cylinder descriptions from contour data. These properties were then used to recover SHGCs in several researches. Sato and Binford [16] described a system for recovery of SHGCs in an edge image. Zerroug and Nevatia [17] presented an approach which addressed the problem of shape description and scene segmentation (figure-ground problem) for SHGCs in the presence of broken contours, markings and occlusion. The approach exploited projective invariant properties of SHGCs in order to guide the segmentation and description processes. The method used these rigorous properties in a bottom up, perceptual grouping approach to handle a hierarchy of three levels of features: curves, symmetries and SHGC surface patches. Gross and Boult [18] proposed a methodology which is useful for the recovery of object classes like tubes, where contour and heuristic constraints are shown to be insufficient for shape recovery. They proved that SHGC contours generated under orthography have exactly two degrees of freedom, and showed that the remaining free parameters can be resolved using reflectance-based (intensity) constraints in addition to contours. The reflectance-based recovery algorithm is demonstrated on both synthetic and real SHGC images.

Ulupinar and Nevatia [19] presented techniques for the recovery of two broader classes of curved objects, namely SHGCs and PRCGCs, and proved that they exhibit certain symmetry properties. The method is based on, and is a major generalization of, the technique they had described for inferring 3D shape from contours of zero-Gaussian curvature (or ZGC) surfaces [20]. In [21], Zerroug and Nevatia extended the approach in [17] to handle PRCGCs and circular PRGCs so as to address more complex, compound, objects which consist of parts of some generic shapes which has both straight and curved axes. They derived a number of rigorous geometric invariant and quasi-invariant properties of these parts and developed a system that automatically detects their projections and recovers their 3D shapes from a 2D image. Havaldar and Medioni [22] addressed the problem of recovering high-level, volumetric and segmented (or part-based) descriptions of objects from three intensity images. They derived a way of inferring volumes in terms of GCs, when, in reality, the unseen parts of the object may be quite different from the inferred ones. In such cases, integrating multiple triplets of images from different positions can provide more accurate volumetric descriptions.

More recently, Zerroug and Nevatia described a method in [23] that detects and describes complex objects (i.e., those containing multiple parts). The method recovered descriptions of single-part objects in terms of three subclasses of GCs (SHGCs, PRCGCs, and circular PRGCs). The method addressed the figure-ground problem and part-based shape description problem at the same time and in a generic way (i.e., without prior knowledge of the specific objects being viewed). It has successfully extracted generic 3D volumetric descriptions composed of certain subclasses of generalized cylinders and of their joint relationships, in the presence of noise, boundary breaks, markings, shadows, and partial occlusion.

Different from the objective of the previous works, in this paper, we will propose a new approach to obtaining a GC-based shape description of 3D objects. The proposed approach of deriving the GC axis is a further extension of the potential-based skeletonization approach presented in [1]. Assume the object surface and its two end cross-sections are given in advance, the GC axis can simply be generated by identifying the potential valley connecting the centroids of them. Cross-sections at different locations along the GC axis are then obtained by a simple algorithm.

The remainder of this article is organized as follows. The generalized potential model in the 3D space is reviewed in the next section. The potential-based skeletonization algorithm is adopted to derive the GC axis of a 3D object in Section 3, where the algorithm for the derivation of cross-sections will also be presented. Section 4 gives some experimental results and discussion of the proposed approach to deriving the potential-based GC representation. Section 5 concludes this chapter and outlines some possible directions for future research.

2. A review of generalized potential fields in the 3D space

In [24], a potential-based modelling of 3D workspace for collision avoidance is proposed. It is shown that the Newtonian potential, being harmonic in the 3D space, can not prevent a point charge from running into an object surface which is uniformly charged. This is because the value of such a potential function is finite at the continuously charged surface. Subsequently, generalized potential models are developed to assure collision avoidance between a point and polyhedral surfaces in the 3D space. The potential function is inversely proportional to the distance between two point charges to the power of an integer (m) and the potential and thus its gradient due to a 3D polygon can be calculated analytically. In particular, it is shown that the repulsive force exerted on a point charge p located at (x, y, z) due to a 3D polygon S can be obtained analytically by evaluating the gradient of the following potential function, for $m = 3$:

$$\Phi(x,y,z) = \sum_i [\phi(x_2^i, y_2^i, z) - \phi(x_1^i, y_1^i, z)] + \frac{\alpha}{z} \tag{1}$$

with

$$\phi(x,y,z) = \frac{1}{z}\tan^{-1}\frac{xz}{y\sqrt{x^2+y^2+z^2}}, \tag{2}$$

where the $x^i - y^i - z$ coordinate system is determined by the right-hand rule for each edge i of S such that z is measured along the normal direction of S and x^i is measured along edge i of S, respectively. As for α, it is the angular extent of the projection of p on S, which is lying inside S. For example, $\alpha = 2\pi$ if the projection is inside S, $\alpha = \pi$ if the projection is on an edge of S, and α is equal to the angle between two edges if the projection is on a vertex of S where the two edges are connected. Hence, the repulsive force exerted at p due to polygon j can be denoted as

$$\overrightarrow{f^j} = (f^j_x, f^j_y, f^j_z) = \left(\frac{\partial\Phi^j}{\partial x}, \frac{\partial\Phi^j}{\partial y}, \frac{\partial\Phi^j}{\partial z}\right). \tag{3}$$

Using the above analytical expressions, the evaluation of the resultant repulsion between a 3D object, a m charged polygonal surfaces, and a seed point can be estimated in closed form through superposition. Thus, the resultant repulsion force can be denoted as

$$\sum_j^m \overrightarrow{f^j} = \left(\sum f^j_x, \sum f^j_y, \sum f^j_z\right). \tag{4}$$

The repulsion will be used to generate the potential-based GC axis for a 3D object, as discussed next.

3. A new generalized cylinder approach

In this section, we will propose a new approach to derive a GC-based shape description of 3D objects based on the generalized potential model reviewed in the previous section. The GC representation consists of a GC axis and a cross-section function defined along this axis. Assume the two ends (cross-sections) of a 3D object are given in advance, the GC axis can be derived by identifying the potential valley connecting the centroid points of them. Cross-sections at different locations along the GC axis are then obtained by a simple algorithm. Thus, given object surface and its two end cross-sections, the proposed approach to deriving its GC representation can be divided into two phases, as shown in Fig. 2. In the first phase, the GC axis is generated using the potential-based skeletonization algorithm by connecting the two seeds. Then, the cross-sections at different locations along the derived GC axis are obtained in the second phase.

The algorithms for the derivation of the GC axis and the associated cross-sections of a 3D object will be presented in Sections 3.1 and 3.2, respectively. In Section 3.3, the possibilities of using alternative seed points in the deriving GC representation are discussed.

3.1. GC axis generation

Since the GC axis usually lies in the middle of an object, which is in a way similar to an object skeleton, the proposed approach adopts the potential-based skeletonization algorithm presented in [1], as reviewed next, to obtain the GC axis. The potential-based skeletonization algorithm identifies the potential valleys connecting several input seed points, which are selected as the convex vertices of the object (i.e., end points of the derived object skeleton), to obtain the object skeleton. The algorithm for generating the GC axis is essentially the same as the skeletonization algorithm except that the input seeds are selected as the centroids of two cross-sections of the object instead the convex vertices. With the boundary of a polyhedron charged according to the generalized potential model described in the previous section, the basic requirement for skeletonization, i.e., the potential will diverge at boundaries of polyhedral objects, is guaranteed. Inside the object boundary, the potential valley corresponds to 3D locations where the potential has locally minimal value in the 2D subspace perpendicular to the direction of the potential gradient. The skeletonization approach proposed in [1] obtained the object skeleton by, starting from selected seed points, traversing the corresponding potential valleys until a potential minimum is reached. The basic computation of the MAT skeleton, starting with one of the seed point, is described in the following algorithm :

Algorithm Potential_Skeleton

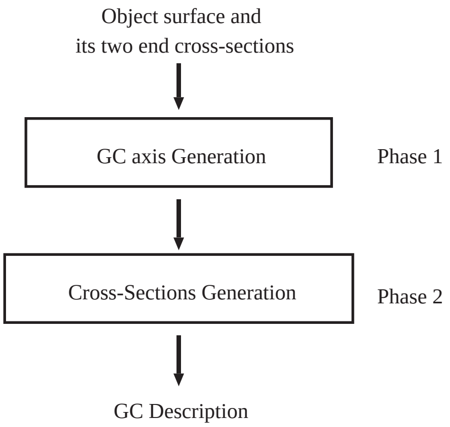


Fig. 2. The two phases for obtaining the GC description of an object.

Step 1: Follow the direction of the force to traverse the skeleton (potential valley) until a zero force is obtained, i.e., a potential minimum is reached.
Step 2: Repeat Step 1 for each of the seed points.
Step 3: End the skeleton computation if there is only one potential minimum.
Step 4: Derive additional skeleton branches by identifying potential valleys connecting neighboring potential minima.

In Step 1, the force following is performed inside the polyhedral boundary. For a closed polyhedron, there exists at least one potential minimum since the potential and thus the force field will diverge at its surface [24]. If the number of potential minima obtained from the above force following procedure is equal to one, the complete skeleton is generated and Step 4 is not necessary. Otherwise, the potential minima need to be connected to form the complete MAT skeleton. To estimate the potential skeleton between these potential minima, we first connect each pair of neighboring minima with a virtual line segment.[2] Equally spaced point samples are then selected along the line segment as initial guesses of point samples along the associated skeleton branch. The potential valley between the two minima is then estimated by searching the minimum potential location in the 2D subspace perpendicular to the virtual line segment at each of these samples.

Fig. 3(a) shows the skeleton of a rectangular hexahedron obtained with *Potential_Skeleton*. The seed

[2] In some complex cases where the line segment intersects the object boundary, a more refined estimation based on the visibility graph method is necessary. See [1] for more detailed discussion.

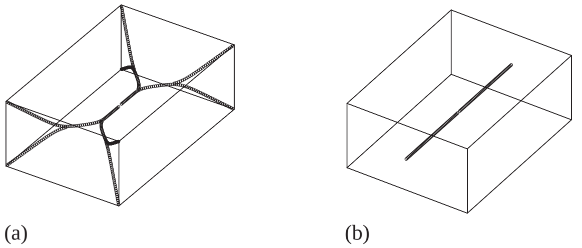


Fig. 3. (a) A rectangular hexahedron and its MAT skeleton. (b) A GC axis of the rectangular hexahedron.

points, where the force following procedure of the algorithm start, are selected as the eight convex corners of the object. (In the computer implementation, the selected seed points are located inside the object boundary and are slightly away from those convex corners.) The derived skeleton is closely related to the media surface derived in [25], as discussed in [1]. Obviously, the derived skeleton differs from the GC axis which is defined in Section 1 as the principal axis in reflecting the topology of an object. However, if we select the seed points as the centroid points of the two ends (cross-sections) of the object instead of the eight corners, a GC axis will be obtained directly with the same algorithm, as shown in Fig. 3(b).[3]

The main difference between the GC axis and the object skeleton in describing the object shape can be stated most conveniently by considering prismatic objects. While the axis and the cross-section of GC will simply be the principal axis and the cross-section of a prism, respectively, the object skeleton obtained with the proposed approach are more complex near the two ends of the prism where the topology of the skeleton is determined through the selection of seeds (which are convex vertices of the cross-sections in our case). In general, for an object consists of elongated parts, both representations will reflect similar topological (skeletal) properties of each part of the object, except for the above differences, and in turn those of the whole object.

In addition, different from the object skeleton, the GC representation is not unique in general. For example, the GC axis can have either horizontal or vertical direction for a rectangle in the 2D space. As shown in Fig. 4(a), we may select points X and Y as two seeds and generate the GC axis $\overline{XY}$. Similarly, points M and N can also be selected as seeds to generate the GC axis in a different direction, as shown in Fig. 4(b). Figs. 4(c) and (d) show the GC cross-sections corresponding to $\overline{XY}$ and $\overline{MN}$, respectively. In Section 3.3, some suggestions for seed point selection will be given from the observation on the different GC axes and cross-sections.

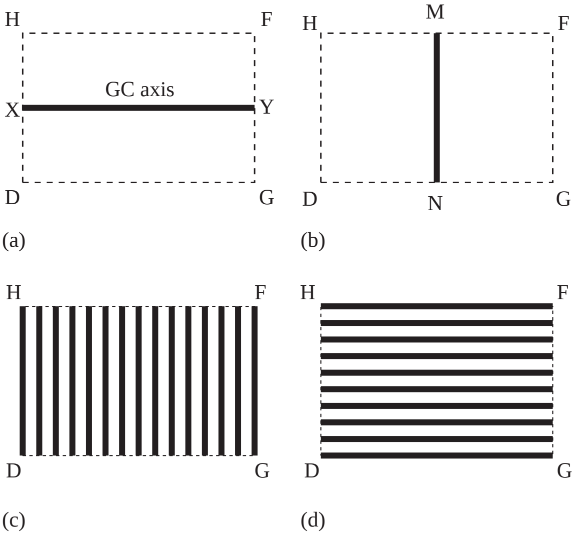


Fig. 4. The GC axis and cross-sections of a rectangle in the 2D space.

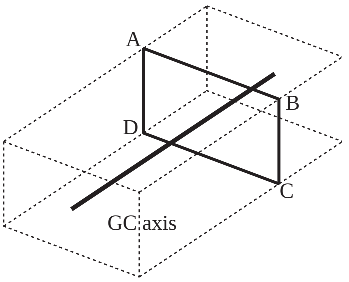


Fig. 5. One GC cross-section $ABCD$ of a rectangular solid.

3.2. Cross-sections generation

The algorithm for obtaining the cross-sections of a generalized cylinder is based on the definition proposed by Binford: a cross-section of an object is called a cross-section of a generalized cylinder if its surface normal is parallel to the GC axis. For example, polygon $ABCD$ shown in Fig. 5 is one of the GC cross-sections of the rectangular solid since the normal of $ABCD$ is parallel to the GC axis.

Given object surface and the derived GC axis, a simple procedure for obtaining the GC cross-sections for the sequence of n sampling points S_i, $1 \leqslant i \leqslant n$, of the derived GC axis can be summarized as follows.

Algorithm Cross_Section_Gen

```
Variables: $S_i$ Sampling Point $i$; $P_i$ Plane $i$; $n$ number of
           sampling points;
Step 1:    Set $i = 1$.
```

[3] Theoretically, the MAT skeleton and GC axis are both continuous curves, however, they are searched step by step in our implementation and then yield a discrete form of MAT skeleton and GC axis.

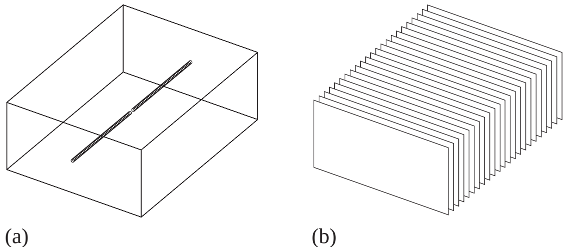


Fig. 6. GC representation of rectangular hexahedron: (a) GC axis, (b) cross-sections.

Step 2: Find the plane P_i which is perpendicular to $\overrightarrow{S_iS_{i+1}}$ at S_i.

Step 3: Compute all intersection points and line segments of P_i and the object surfaces.

Step 4: Connect these intersections sequentially to form a polygon, i.e., one cross-section of the object.

Step 5: $i \leftarrow i+1$, repeat Step 3 if $i \leqslant n$.

Step 6: End of the algorithm.

Fig. 6(a) shows the GC axis of a rectangular hexahedron obtained by using *Potential_Skeleton* in Phase 1. The two centroids of the two ends of the object are selected as seeds. Based on the derived GC axis, a complete shape description of the object is then obtained in Phase 2 using *Cross_Section_Gen*, as shown in Fig. 6(b). Obviously, the selection of seed points, will greatly affect the result of the derived GC axis and cross-sections. A more detailed discussion about this issue will be given in the next subsection.

3.3. Selection of seed points

In this subsection, we will discuss the influence of seed point selection on the derived GC axis and cross-sections. Consider the rectangular hexahedron shown in Fig. 7. The MAT skeleton of the hexahedron obtained by *Potential_Skeleton* using eight convex corners as seed points is shown in Fig. 7(a). As for the GC axis of the object, Fig. 7(b) shows the axis obtained using two of the eight corner points as seeds whereas centroids of the front and back end surfaces of the object are selected for the axis shown in Fig. 7(c). The corresponding cross-sections obtained by *Cross_Section_Gen* for Figs. 7(b) and (c) are shown in Figs. 7(d) and (e), respectively.

Obviously, the cross-sections shown in Fig. 7(d) are too complex due to the large curvature of the GC axis at two locations, as shown in Fig. 7(b). On the other hand, the GC axis shown in Fig. 7(c) goes straight from the front to the back end of the object while maintaining maximum distances from the other four sides of the object. Thus, the cross-sections derived for the GC axis shown in Fig. 7(c) are much simpler (all of identical shape and size) and more appropriate as the representation of the rectangular solid.

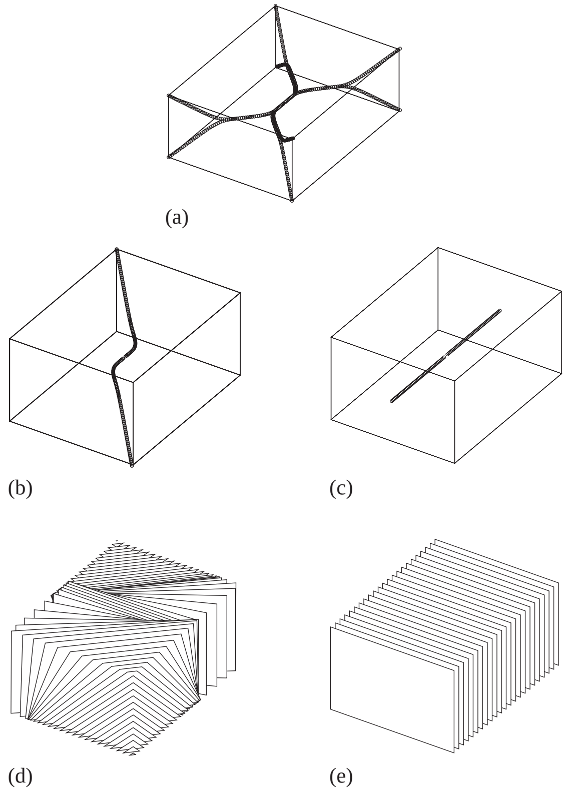


Fig. 7. The MAT skeleton and different GC representations of a rectangular hexahedron (see text).

According to the above discussion, one can see clearly that, the selection of seed points plays a crucial role in the GC algorithm proposed in this paper. In general, seed points should be selected such that a GC axis with smaller curvature can be generated and the corresponding cross-sections will be simpler and neater for representing the object. This is especially the case for elongated object with cross-sections of identical size and shape. However, a systematic way of identifying optimal seed points for an object of arbitrary shape is yet to be established.

4. Experimental results and discussion

In this section, we will show simulation results for generating the potential-based GC axes and cross-sections of simple as well as more complex objects using the algorithm presented in the previous section. In each of the examples, the seed points used in the force following procedure are assumed to be given in advance that an appropriate GC representation of each object can be generated. In general, it takes more computation

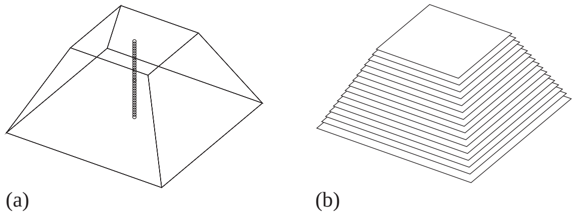


Fig. 8. GC representation of echelon hexahedron: (a) GC axis, (b) cross-sections.

time to derive the GC representation for more complex objects. In some cases, for objects containing multiple parts, a decomposition of an object into some simple parts is needed for the proposed approach to work properly. The algorithm is implemented with the C programming language and all examples are tested on a Sun Sparc Ultra-1 workstation with SunOS 4.1.4 and 128MB Main Memory.

4.1. Simple models

In this subsection, the proposed approach is applied to obtain the GC representations of objects of simple shapes. For each object, it is assumed that the two end cross-sections, as well as the two ends of the GC axis, are given in advance. Accordingly, the latter are used as seed points in the force following procedure of *Potential_Skeleton* in each example.

Fig. 8(a) shows the GC representation obtained with the proposed approach for an echelon hexahedron. Since only one potential minimum is obtained with Step 1 of *Potential_Skeleton*, the connecting process given in Step 4 is not needed. Due to the symmetry of the object shape, the GC axis corresponds to a line segment. Similar results can be obtained for the potential-based GC representations shown in Figs. 9 and 10. The computation times for the above examples are summarized in Table 1.[4] Note that Figs. 6 and 8 give the SHGC representation (but not Figs. 9 and 10) since there is no change in shape for the cross-sections. In Fig. 10, two object vertices, instead of face centers, are used as seed points, i.e., the area of the two end cross-sections of its GC representation is equal to zero.

The 3D objects considered so far have only one potential minimum in their interior; therefore, additional connecting process given in Step 4 of *Potential_ Skeleton* is not required. Figs. 11 and 12 show two examples in which more than one potential minimum can be found. In Fig. 11(a) (Fig. 12(a)), distance between the two potential minima obtained with Step 1 of

[4]For the examples considered in this section, only a subset of the total number of cross-sections calculated are shown in the corresponding figures for better illustrations.

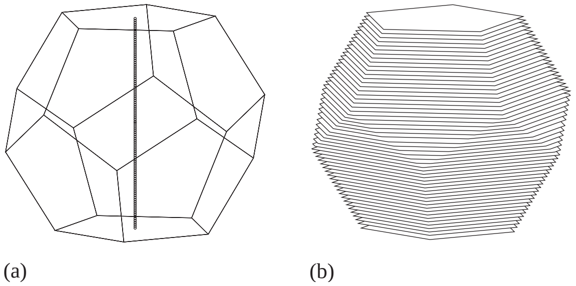


Fig. 9. GC representation of dodecahedron: (a) GC axis, (b) cross-sections.

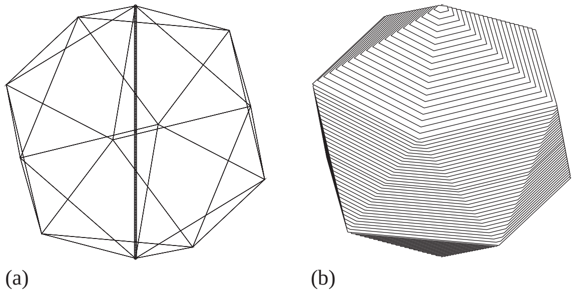


Fig. 10. GC representation of icosahedron: (a) GC axis, (b) cross-sections.

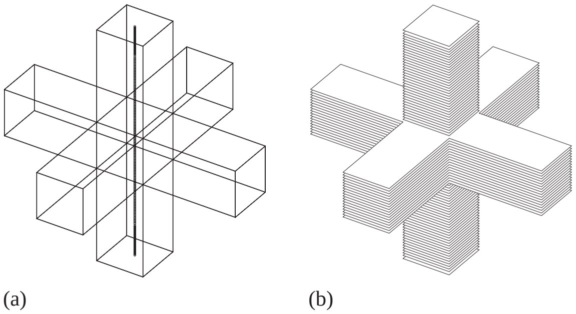


Fig. 11. GC representation of 3D polyhedron with three long rectangular polyhedra crossed over: (a) GC axis, (b) cross-sections.

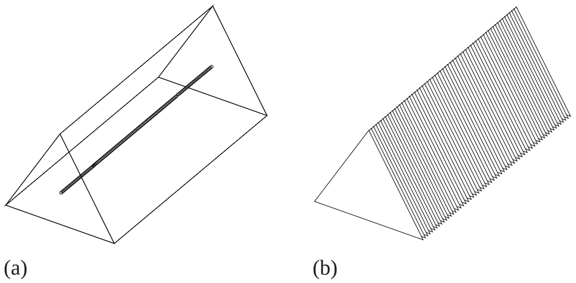


Fig. 12. GC representation of tetrahedron: (a) GC axis, (b) cross-sections.

Table 1
Computation times for different GC representations

Process	Objects	Vertices	Polygons	Total Numbers		CPU Time (s)	
				S_is	CSs	GC axis	CS
Connection not required	Rectangular hexahedron	8	6	94	47	0.098	0.08
	Echelon hexahedron	8	6	34	17	0.071	0.028
	Dodecahedron	20	12	114	57	0.51	0.146
	Icosahedron	14	20	156	78	0.786	0.344
	Adapter	79	50	32	16	0.569	0.116
	Cap	639	386	30	15	3.54	0.895
	Ball2	994	1024	62	31	16.207	1.681
	Propane	128	66	110	55	2.333	1.281
	SHGC	650	1238	250	125	64.357	7.619
	PRCGC	456	908	194	97	38.565	4.011
	Circular PRGC	456	908	194	97	36.708	4.057
Connection required	3D cross	24	30	178	89	0.981	0.269
	Tetrahedron	6	5	128	64	12.122	0.108
	Column1	392	262	112	56	45.038	2.063

Potential_Skeleton is equal to 3/4 (slightly less than 3/5) the length of the GC axis. Due to additional connecting process (Step 4 of *Potential_Skeleton*) required in generating the GC axis, more computation time (see Table (1) is needed. More in depth analysis of computational complexity will be discussed later in Section 4.3. As for the cross-sections, since there is no restriction on the changes of their size, shape, etc., the changes may not be continuous, e.g., in Fig. 11.

4.2. Complex models

In this subsection, the object models are obtained from a database of 3D objects, which can be found at the website http://sampl.eng.ohiostate.edu/~s-ampl/data/3DDB/Models/. Each object in the database is stored in a file format specifying its vertices, polygons, and surfaces. The main difference between simple and complex object models is the latter use a lot of polygons to describe an object surface such that the object looks more real.

Figs. 13–16 show experimental results of the proposed approach for some objects represented by complex models wherein one potential minimum is obtained with *Potential_Skeleton* for each example. The computation times for the above examples are summarized in Table 1. Due to more complex boundary description, these computation times are in general longer than those for the simple models. Fig. 17 shows another example which requires additional connecting process to connect the two potential minima while deriving the GC axis. It is obvious that the GC representation shown in Fig. 17 can be regarded as two simple SHGCs connected back-to-back. (Similar observations can be made for Fig. 11 but not for Figs. 9 and 10.) In this case, the axes of the two SHGC are connected into a single line segment.

In general, for an object with complex geometry, decomposing the object into simpler parts then obtaining the GC representation for each part may be necessary. One way of obtaining the GC representation

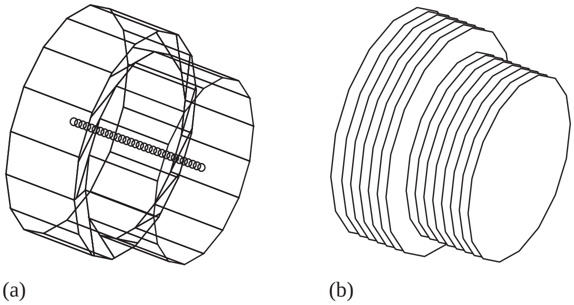


Fig. 13. GC representation of adapter: (a) GC axis, (b) cross-sections.

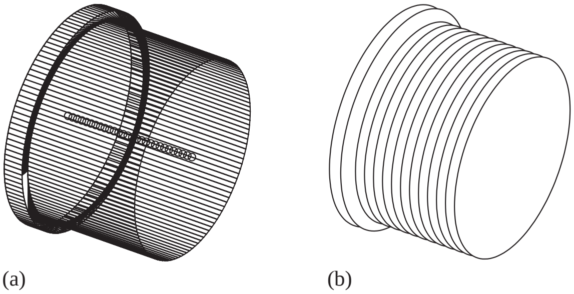


Fig. 14. GC representation of cap: (a) GC axis, (b) cross-sections.

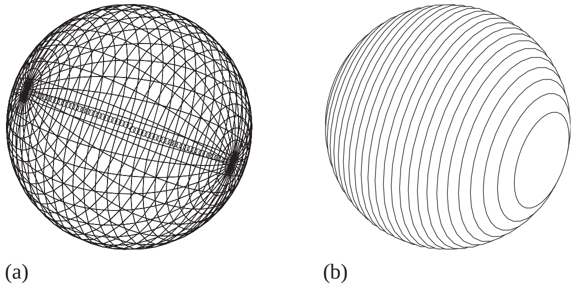


Fig. 15. GC representation of ball2: (a) GC axis, (b) cross-sections.

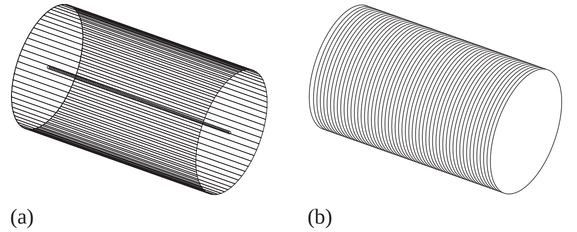


Fig. 16. GC representation of propane: (a) GC axis, (b) cross-sections.

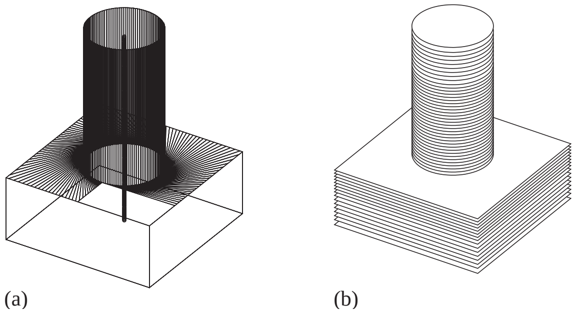


Fig. 17. GC representation of column1: (a) GC axis, (b) cross-sections.

for such a complex shaped object is to decompose the object into simpler parts after the GC axis of the complete object is derived, Then, cross-sections of each part of the object are obtained. For example, Fig. 18 shows a Y-shaped object composed of two pipes: a slim and short pipe and a wider and longer one. First, the GC axis can be derived by selecting three seed points[5] for *Potential_Skeleton*. The connecting process is not required since the force following procedure yields an unique potential minimum, as shown at the top of Fig. 18. The resultant GC axis is generated within 10.256 s. Secondly, the object as well as its GC axis (the separated GC axes are not shown in the figure) are divided into two parts. Thirdly, the two parts are processed separately to derive their cross-sections using *Cross_Section_Gen*. Finally, the entire GC representation of the object is obtained by merging the two GC representations, as shown at the bottom of Fig. 18. The computation time spent on obtaining the 22 cross-sections of the left part is 0.256 s, while the right part takes 0.114 s to generate 9 cross-sections.

In describing the simple parts of a complex shaped object, some important subclasses of GCs such as SHGC, PRCGC, and Circular PRGC are commonly used. Examples of these three subclasses of GCs are shown in Fig. 1. In the following examples, some of such models are created by 3D Studio Max to test the proposed approach. Figs. 19(a) and (b) show the GC axis and cross-sections of a SHGC, respectively, generated by the proposed approach. Similar results are shown in Figs. 20 and 21 for a PRCGC and a circular PRGC, respectively.

4.3. Computation complexity

The proposed approach is not only easy to implement, the associated algorithm is also of low-time complexity because the generalized potential field is analytically tractable. Suppose a polyhedron is composed of n polygons. For each step of the force following procedure, the complexity for calculating the repulsive force exerted on a point sample on the GC axis due to those polygons is $O(n)$. Thus, for obtaining m' point samples of the GC axis through force following, the time complexity is $O(nm')$. Similarly, it is not hard to show that if m'' points are needed to connect the potential minima obtained from the force following procedure, the time complexity for finding these m'' samples of the GC axis is $O(nm'')$. Thus, the time complexity for finding the complete GC axis is $O(nm)$, where $m = m' + m''$ is equal to the total number of point samples of the GC axis. Therefore, the computation time for deriving the GC axis is approximately a linear function of the product of the number of object polygons and that of point samples of the GC axis. For example, Fig. 22 shows the computation time for deriving the GC axis of experimental examples listed in Table 1 for which the connection process is not required. The additional points marked by 'X' are derived from the same SHGC but different sample points. As for the generation of cross-sections, since the computational complexity highly depend on the shape of an object as well as the distribution of surface polygons, its analysis is not straightforward in general.

[5] It is readily observable that the resultant GC representation is not good enough if we select only two seed points. Therefore, three seed points are manually identified corresponding to three end cross-sections (circular faces) in this case.

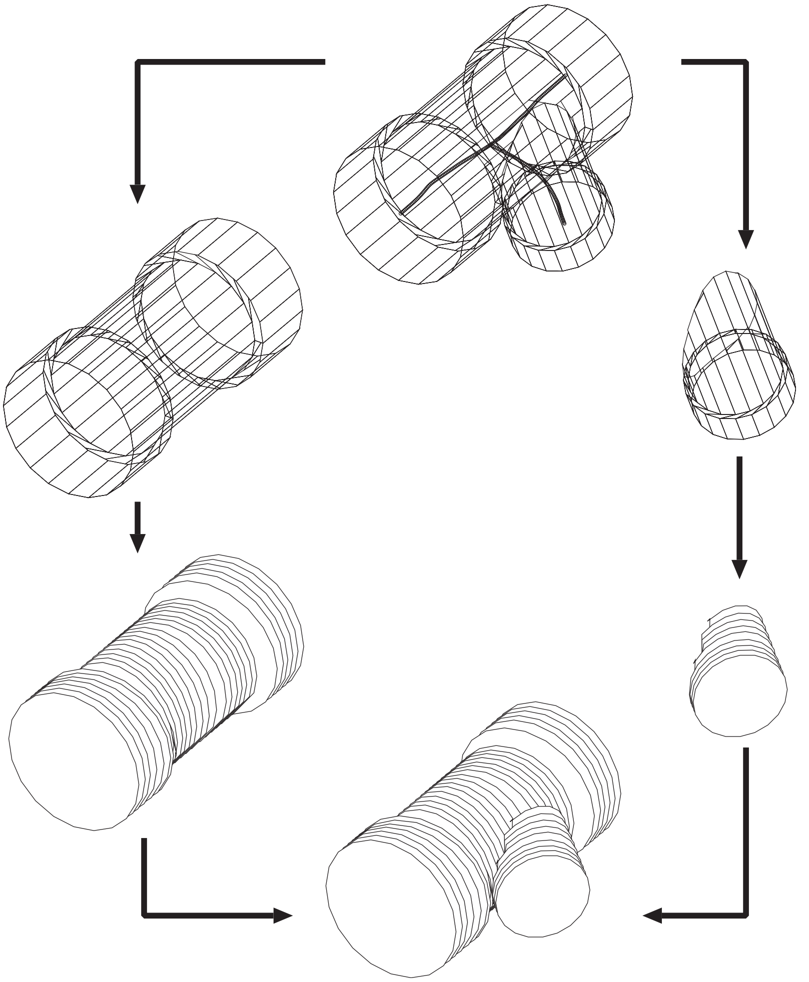
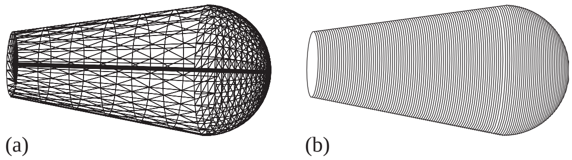
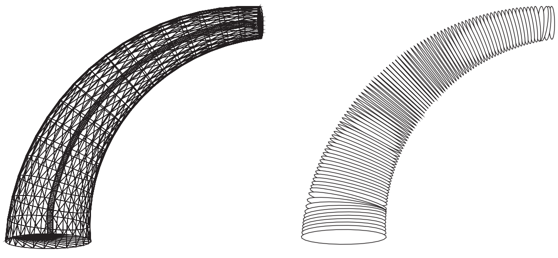


Fig. 18. Generating the GC representation of a complex shaped object which need to be divided into two parts and processed separately.



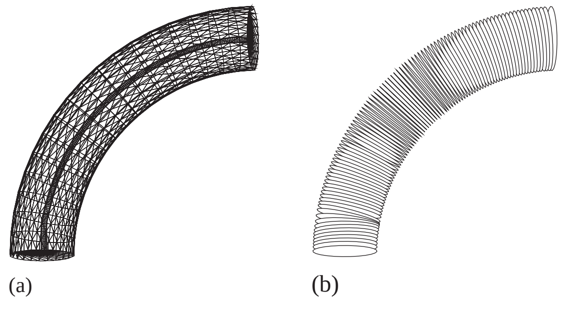
(a) (b)

Fig. 19. SHGC (a) GC axis, (b) cross-sections.



(a) (b)

Fig. 21. Circular PRGC (a) GC axis, (b) cross-sections.



(a) (b)

Fig. 20. PRCGC (a) GC axis, (b) cross-sections.

4.4. Alternative surface description for GC derivation

In the previous section, the approach of GC axis generation uses two seed points, the two ends of the axis, for *Potential_Skeleton*. In some cases, an alternative surface description may be adopted for more efficient

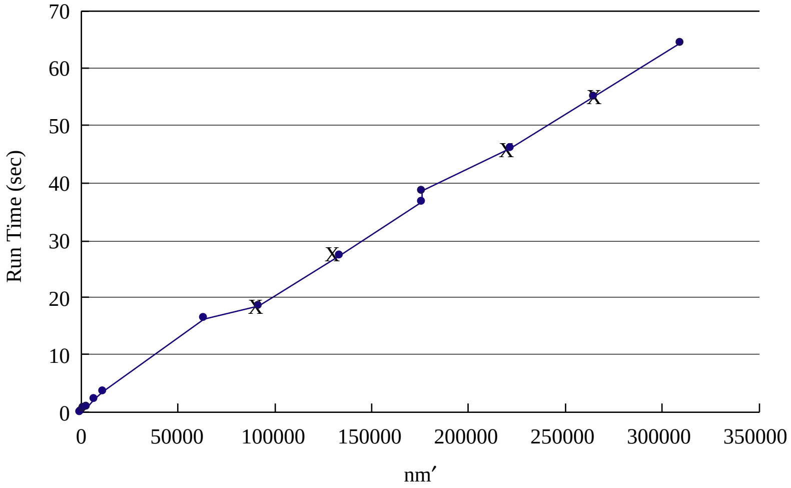


Fig. 22. The computation time of axis generation for some GC representations.

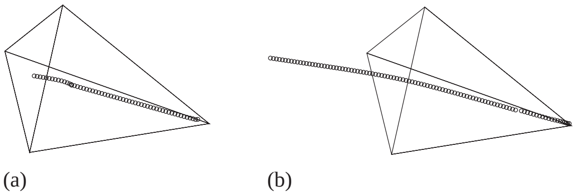


Fig. 23. (a) A tetrahedron and its GC axis generated using two seed points, (b) the trajectory of a seed point for an deleted end surface.

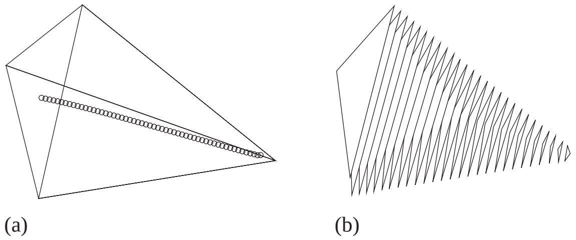


Fig. 24. GC representation of the tetrahedron obtained by using an alternative surface description: (a) GC axis, (b) cross-sections.

GC derivations. Fig. 23(a) shows a tetrahedron and its GC axis derived using two seed points, a vertex and the centroid of an end surface. Alternatively, if one deletes the end surface and use only the above vertex as seed point for *Potential_Skeleton*, the seed point will move outside the object region as shown in Fig. 23(b). Fig. 24 shows the resultant GC axis, which corresponds to the trajectory of the seed point inside the tetrahedron, and the associated GC cross-sections. The computation time spent on deriving the GC axes of the tetrahedron for 54 axis points are equal to 0.084 and 0.063 s, respectively, for Figs. 23(a) and 24(a).

5. Conclusions and future works

In this paper, we have proposed a new approach to derive a GC-based shape description of 3D objects using the generalized potential model. The proposed approach of deriving the GC axis is an extension of the potential-based skeletonization approach introduced in [1]. The GC representation which consists of this GC axis and corresponding cross-sections can be generated automatically for the two ends of the axis given as inputs. In general, the derived GC representations will yield better approximation of object shape than that based on simpler subclasses of GC since there is, in principle, no restriction on the topology of the GC axis and the shape of the cross-sections. On the other hand, any simpler subclass of GC which has a simpler GC topology may also be obtained from the potential-based GC representation through appropriate simplifications, in cross-section/axis descriptions, of the latter.

References

[1] Chuang J-H, Tsai C-H, Ko M-C. Skeletonization of three-dimensional object using generalized potential field. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000;22(11):1241–51.

[2] Binford TO. Visual perception by computer. Proceedings of the IEEE Conference on Systems and Controls. Miami, FL, December 1971.

[3] Nevatia R, Binford TO. Description and recognition of complex curved objects. Artificial Intelligence 1977;8(1): 77–98.

[4] Marr D, Nishihara HK. Representation and recognition of the spatial organization of three-dimensional shapes. Proceedings of the Royal Society of London 1978; B-200:269–94.

[5] Pentland A. Recognition by parts. Proceedings of the International Conference on Computer Vision, ICCV, 1987. p. 612–20.

[6] Shafer SA, Kanade T. The theory of straight homogeneous generalized cylinders. Technical Report CS-083-105, Carnegie Mellon University, 1983.

[7] Shafer SA. Shadows and silhouettes in computer vision. New York: Kluwer Academic Publishers; 1985.

[8] Agin GJ. Representation and description of curved objects. Ph.D. thesis. Stanford Artificial Intelligence Laboratory, Report AIM-173, October 1972.

[9] Marr D. Analysis of occluding contour. Proceedings of the Royal Society of London 1977;B-197:441–75.

[10] Horaud R, Brady JM. On the geometric interpretation of image contours. Proceedings of the First International Conference on Computer Vision, London, UK, June 1987.

[11] Nalwa V. Line-drawing interpretation: bilateral symmetry. IEEE Transactions on Pattern Analysis and Machine Intelligence 1989;11:1117–20.

[12] Richetin M, Dhome M, LaPeste JT, Rives G. Inverse perspective transform using zero-curvature contours points: applications to the localization of some generalized cylinders from a single view. IEEE Transactions on Pattern Analysis and Machine Intelligence 1991;13(2):185–92.

[13] LaVest JM, Glachet R, Dhome M, LaPreste JT. Modeling solids of revolution by monocular vision. Proceedings of the 1991 CVPR, Lahaina, Hawaii, June, 1991. p. 690–1.

[14] Rao K, Nevatia R. From sparse 3-D data directly to volumetric shape description. Proceedings of the Image Understanding Workshop, Los Angeles, CA, February 1987.

[15] Ponce J, Chelberg D, Mann WB. Invariant properties of straight homogeneous generalized cylinders and their contours. IEEE Transactions on Pattern Analysis and Machine Intelligence 1989;11(9):951–66.

[16] Sato H, Binford TO. Finding and recovering SHGC objects in an edge image. Computer Vision Graphics and Image Processing 1993;57(3):346–56.

[17] Zerroug M, Nevatia R. Volumetric descriptions from single intensity image. International Journal of Computer Vision, 1996; 11–42.

[18] Gross AD, Boult TE. Recovery of SHGCs from a single intensity view. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996;18(2):161–80.

[19] Ulupinar F, Nevatia R. Shape from contour: straight homogeneous generalized cylinders and constant cross section generalized cylinders. IEEE Transactions on Pattern Analysis and Machine Intelligence 1995;17(2): 120–35.

[20] Ulupinar F, Nevatia R. Perception of 3D surfaces from 2D contours. IEEE Transactions on Pattern Analysis and Machine Intelligence 1993;15(1):3–18.

[21] Zerroug M, Nevatia R. Three-dimensional descriptions based on the analysis of the invariant and quasi-invariant properties of some curved-axis generalized cylinders. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996;18(3):237–53.

[22] Havaldar P, Medioni G. Full volumetric descriptions from three intensity images. IEEE Transactions on Pattern Analysis and Machine Intelligence 1998;20(5).

[23] Zerroug M, Nevatia R. Part-based 3D descriptions of complex objects from a single image. IEEE Transactions on Pattern Analysis and Machine Intelligence 1999;21(9).

[24] Chuang J-H. Potential-based modeling of three-dimensional workspace for obstacle avoidance. IEEE Transactions on Robotics and Automation 1998;14(5):778–85.

[25] Sherbrooke EC, Patrikalakis NM, Brisson E. An algorithm for the medial axis transform of 3D polyhedral solids. IEEE Transactions on Visualization and Computer Graphics 1996;2(1):44–61.