An Approximative Calculation of Relative Convex Hulls for Surface Area Estimation of 3D Digital Objects

Linjiang Yu and Reinhard Klette

Abstract

Relative convex hulls have been suggested for multigrid-convergent surface area estimation. Besides the existence of a convergence theorem there is no efficient algorithmic solution so far for calculating 3D relative convex hulls. This article discusses an approximative solution based on minimum-length polygon calculations. It is illustrated that this approximative calculation also proves (experimentally) to provide a multigrid convergent measurement.

Keywords: digital geometry, surface measurement, global polyhedrization, relative convex hull

1. Introduction

Surface area is one of the important three-dimensional (3D) shape features in computer-based image analysis. Recently many papers on theoretical and algorithmic aspects of surface area estimation for 3D digital objects have been published. Polyhedrization approaches are often used for approximation of surface area, and these are also efficient ways to visualize a surface.

The multigrid convergence problem for surface area estimation for 3D digital objects may be stated as follows [1]: assume a measurable solid in 3D Euclidean space being digitized with respect to finer and finer grid resolution. The resulting digital object is used as input for a surface area estimation program; calculated estimates should converge to a fixed value for finer and finer grid resolution, and this fixed value should be the true surface area.

The paper [2] proposed a way to classify polyhedrization techniques and algorithms. It expresses a general hypothesis that local polyhedrization techniques such as surface tracking (opaque cubes), marching cubes, etc. are failing to meet both convergence properties, and global polyhedrization techniques such as convex hull computation (for digitization of convex sets), digital planar segmentations [6], etc. are likely to meet both convergence properties.

J. Sklansky and D. F. Kibler [5] proposed in 1976 the notion of a relative convex hull (RCH in brief), and the multigrid convergence of surface estimations of 3D digital objects based on relative convex hulls is ensured by Theorem 1 in [3]. However, the algorithmic treatment of 3D relative convex hulls remains to be an open problem yet.

For digital objects in 2D space, the minimum length polygon (MLP) is uniquely defined, and the length estimation of digital curves can meet both multigrid convergence constraints using the MLP method (see, e.g. the paper [4] for a linear-time MLP algorithm).

This paper suggests an approximative solution for RCH calculations based on MLP calculations by slicing surfaces of 3D digital objects into 2D digital curves, calculating MLPs in 2D for each of the digital curves, then accumulating the surface area by connecting vertices of MLPs of the digital curves in a certain manner. The class of input sets needs to be restricted for this approximative estimation procedure.

2. Relative Convex Hull (RCH) Approximation

Assume an orthogonal Cartesian coordinate system \( X - Y - Z \). Let \( \Theta \subseteq \mathbb{R}^3 \) be a bounded set. The digital space \( \mathbb{Z}^3 \) is represented using \( r \)-grid cubes \( C_{i,j,k}^r \), where \((i, j, k)\) is the centroid of \( C_{i,j,k}^r \), with six \( r \)-faces parallel to the coordinate planes, with \( r \)-edges of length \( 1/r \), for resolution parameter \( r \geq 1 \) and integers \( i, j, k \) with \( 0 \leq i < \text{dim}_x \), \( 0 \leq j < \text{dim}_y \), and \( 0 \leq k < \text{dim}_z \). We consider a digital object \( J^{-r}(\Theta) \) in \( \mathbb{Z}^3 \) defined by the Jordan inner digitization which consists of all \( r \)-cubes completely contained in the interior of \( \Theta \). The surface area \( S = \partial \Theta \) of \( \Theta \) will be approximated based on \( J^{-r}(\Theta) \) (i.e. a given digital object is assumed to be a set of this type).

Definition 1 A border \( r \)-face \( F_{i,j} \) at position \((i, j)\) in \( +Z \) axis direction is a shared \( r \)-face between two \( r \)-cubes \( C_{i,j,k}^r \) and \( C_{i,j,k+1}^r \) where \( C_{i,j,k}^r \) is in \( J^{-r}(\Theta) \), and \( k + 1 = \text{dim}_z \) or \( C_{i,j,k+1}^r \) is not in \( J^{-r}(\Theta) \). The position \((i, j)\) represents a pair of values on \( X \) and \( Y \) coordinates with \( 0 \leq i < \text{dim}_x \) and \( 0 \leq j < \text{dim}_y \).
Similar border faces can be defined for $-Z$, $+Y$, $-Y$, $+X$, or $-X$ axis directions. According to this definition, it is possible to have zero or more border $r$-faces along position $(i, j)$ in $+Z$ axis direction for a surface of a digital set $J^-_r(\Theta)$, e.g. an object that possesses cavities may have more than one border $r$-face along a certain position. Each border $r$-face $F_{i,j}$ is defined by four corner points $P_0, P_1, P_2, P_3$ if $F_{i,j}$ exists; otherwise let $F_{i,j} = \emptyset$. The surface of a digital set $J^-_r(\Theta)$ can be partitioned into six disjoint axial manifolds in their corresponding axial directions, which might be indexed with $+Z$, $-Z$, $+Y$, $-Y$, $+X$, and $-X$.

**Definition 2** A surface of a digital set $J^-_r(\Theta)$ is called an orthogonally completely visible surface if it has at least one border $r$-face along any position, for each of the six axial manifolds.

Our algorithm is only applicable to orthogonally completely visible surfaces. All digitally convex objects and some non-convex objects possess orthogonally completely visible surfaces. Assume that a digital set $J^-_r(\Theta)$ possesses an orthogonally completely visible surface, the estimated surface area is the sum of all areas of all of these six axial manifolds. For the sake of describing our approximative RCH algorithm simply and clearly, we will explain the area estimation for the $+Z$-axial manifold, and processes are similar for the other axial manifolds. The 8-neighborhood of a border $r$-face $F_{i,j}$ is defined as follows:

**Definition 3** The 8-neighborhood of a border $r$-face $F_{i,j}$ is the set $N_8((i, j))$ of border $r$-faces $F_{x,y}$ such that $\max\{|x - i|, |y - j|\} = 1$.

Each corner point $P_\epsilon (0 \leq \epsilon \leq 3)$ of a border $r$-face $F_{i,j}$ may have a maximum of three further corresponding corner points on $N_8((i, j))$ such that they have the same values on $X$ and $Y$ axes. We define these corner points as corresponding corner points of $P_\epsilon$ on $F_{i,j}$.

Figure 1. Example of MLP calculation for a digital set $J^-_r(\Theta)$ in 2D space using the linear-time MLP algorithm [4].

The digital space $\mathbb{Z}^3$ is divided into $dim_y$ slices along the $Y$-axis. Each slice $\Pi_y$, for $0 \leq y < dim_y$, consists of all grid cubes $C_{y,k}^i$, with $0 \leq i < dim_x$ and $0 \leq k < dim_z$. Two border $r$-faces are edge-connected if they are 8-neighbors.

**Definition 4** A face run is an edge-connected component of border $r$-faces within a slice of an axial manifold in $\mathbb{Z}^3$.

A slice $\Pi_y$ may contain zero or more face runs, and a single face run may consist of just one $r$-face. Each face run in the $+Z$-axial manifold defines two 4-connected vertex sequences (one on each side of the face run) in the $X-Z$ plane specifying sequences of potential vertices of an MLP. (Figure 1 illustrates an example of MLP calculation in the paper [4]) approximating this face run. Each vertex of the 4-connected vertex sequences is a corner point of a border $r$-face by replacing all $Z$ value by the maximum $Z$ value among all corresponding corner points and itself.

Our approximative relative convex hull (RCH) algorithm consists of the following steps for a $+Z$ axial manifold:

1. The digital space $\mathbb{Z}^3$ is divided into $dim_y$ slices along the $Y$-axis.
2. For each slice in $\mathbb{Z}^3$, if there exists any face run, go to the next step. Otherwise, do nothing for this slice.
3. For each face run, obtain both 4-connected vertex sequences in the $X-Z$ plane which trace through potential vertices of the MLP.
4. Use the 2D MLP algorithm to calculate an MLP segment for these two sequences. Then triangulate the resulting MLP vertices of the MLP segment and accumulate all triangle areas into a resulting surface area value for the $+Z$ axial manifold.
5. If all slices are finished, return the surface area value for the $+Z$ axial manifold.

Figure 2. Example of triangulation of a MLP segment for a $+Z$ axial manifold.
Now let us discuss the case of the triangulation of an MLP segment for a +Z axial manifold. Let \((P_0^{(1)}, P_1^{(1)}, \ldots, P_m^{(1)})\) and \((P_0^{(2)}, P_1^{(2)}, \ldots, P_m^{(2)})\) be the vertex sequences of two MLPs, i.e. MLP_1 and MLP_2, along +X direction. Two pairs of points \((P_1, Q_1)\) and \((P_2, Q_2)\) taken from vertices of MLP_1 and MLP_2 are used to calculate an area between them by triangulation. Let \(P_k^{(1)}=\langle x_k^{(1)}, y_k^{(1)}, z_k^{(1)} \rangle\) represents a vertex of another MLP where \(x_k^{(1)}, y_k^{(1)}, z_k^{(1)}\) are the values of X, Y, Z coordinates of the vertex \(P_k^{(1)}\) respectively, and \(P_k^{(2)}=\langle x_k^{(2)}, y_k^{(2)}, z_k^{(2)} \rangle\) represents a vertex of another MLP where \(x_k^{(2)}, y_k^{(2)}, z_k^{(2)}\) are the values of X, Y, Z coordinates of the vertex \(P_k^{(2)}\) respectively. The process of triangulation will be done in order, one after another, on the two vertex sequences. Let \(P_1=P_1^{(1)}, Q_1=P_2^{(2)}\); the next pair \(P_2, Q_2\) depends on the following situations:

1. \(P_2 = P_1, Q_2 = P_{j+1}^{(2)}\), if \(x_{j+1}^{(2)} > x_{i}^{(1)}\).
2. \(P_2 = P_{i+1}^{(1)}, Q_2 = Q_1\), if \(x_{i+1}^{(1)} > x_{j+2}^{(2)}\).
3. \(P_2 = P_{i+1}^{(1)}, Q_2 = P_{j+1}^{(2)}\), otherwise.

An example of a triangulation of an MLP segment for a +Z axial manifold is demonstrated in Figure 2. Consider a face run of the middle slice \(\Pi_y\), square marks and dot marks represent the vertex sequences of two MLPs, i.e. MLP_1 and MLP_2 respectively. The resulting triangulation consists of the black lines between vertices of the two MLPs, and the sum of areas of all the triangles is the area of the face run.

Note that a face run can be a very long sequence of border r-faces if the grid resolution \(r\) goes to infinity. Our algorithm uses a sliding window on each slice limiting the length of face runs according to memory limitations in the program. This allows to run this algorithm on digital sets of any resolution \(r\). The window size however can be any integer specifying our algorithm as being a global polyhedrization algorithm. The computational complexity in this algorithm is \(O(N)\) where \(N\) is the number of faces of the digital object under grid resolution \(r\).

### 3. Experimental Results

General ellipsoids with semi-axes \(a, b, c\), cuboids and a non-convex object are used as test objects. The relative error \(E_{\text{rel}} = |S_o - S_f|/S_o\) is used to analyze and evaluate the convergence of our polyhedrization algorithm, where \(S_o\) is the surface area estimation of \(\Theta\) and \(S_f\) the true value of surface area of \(\partial \Theta\).

Figure 3 illustrates the surface area estimation of an ellipsoid with semi-axes 20 × 16 × 12 (axes parallel to coordinate axes) if different widths \(w\) are used for the sliding window in our RCH algorithm. The widths \(w = 100, 200, 300, 400\) have been used in this experiment, and results show that all curves illustrate similar multigrid convergence behaviour. Variations in width \(w\) of the moving window do not make a big difference for this algorithm. A reason may be that the resolution (or the size of the object) is not yet large enough with respect to the window size.

Figure 4 illustrates the impact of different orientations of the same ellipsoid on curves of relative errors of estimated surface areas versus grid resolution: rotating 45° about Z-axis, rotating 45° about Z-axis and \(Y\)-axis, and no rotation. It obviously shows that all error curves match both multigrid convergence constraints, and variations of object orientations do not make a difference for the estimated surface area using our algorithm.

Figure 5 illustrates the impact of different object shapes on the speed of convergence. These curves show relative errors of estimated surface areas within a family of different objects, for four different grid resolutions. The family of objects are ellipsoids with semi-axes 20 × 20 × t in orientation parallel to the coordinate axes, where parameter \(t\) is the thickness of the ellipsoid ranged from 2 to 20, i.e. from...
Figure 5. Relative errors of surface area estimation versus a family of objects in different grid resolution.

Finally, a curve of relative errors of estimated surface area versus grid resolution for a non-convex object is illustrated in Figure 6. The test object is a composition of two blocks of frustum of a right circular cone connecting together with their small circular bases. The size of the frustum of the right circular cone is: $\rho=5$, $R=10$, $h=20$. This curve shows a trend to converge towards the true value of surface area, however with some oscillation.

4. Comparison and Conclusions

Figure 7 illustrates the comparison of experimental results using different global polyhedrization algorithms for an ellipsoid with semi-axes $20 \times 16 \times 12$. Figure 8 summarizes the differences on multigrid convergent behavior, time complexity, and applicability among the three global polyhedrization algorithms for surface area estimation. It shows that the time complexity for the RCH algorithm is better than the other two. We conclude that our approximative calculation of relative convex hulls is a possible approach for surface area estimation. However, the object needs to have an orthogonally completely visible surface.

5. Acknowledgement

Thanks to Garry Tee (Auckland University) for providing a program to calculate the surface area of an arbitrary ellipsoid.

References


