Marching Triangles: Delaunay Implicit Surface Triangulation

A.Hilton[1] and J.Illingworth

Vision, Speech and Signal Processing Group,
Department of Electronic and Electrical Engineering
University of Surrey, Guildford. GU2 5XH. U.K.
a.hilton@surrey.ac.uk

Abstract

A new *surface-based* approach to triangulation of an implicit surface called 'Marching Triangles' (MT) is introduced in this paper. MT enables reconstruction of an efficient triangular mesh representation of an open manifold implicit surface of arbitrary topology. The surface-based approach polygonises the implicit surface by growing a triangulated mesh according to the local geometry and topology. A local *3D Delaunay Surface Constraint* is introduced to ensure that the triangulated mesh is locally Delaunay and is a correct surface approximation of the manifold surface. The resulting triangulation is globally Delaunay with uniform triangle shape providing an efficient representation of the implicit surface.

MT overcomes several limitations of previous *volume-based* implicit surface polygonisation techniques. Volume-based approaches, such as Marching Cubes (MC), intersect the surface with a volumetric decomposition of the space. This results in highly inefficient representation due to non-uniform triangle shape. MT also enables polygonisation of open manifolds, dynamic integration of new data into an existing triangulation, reduced computational cost and correct approximation of complex geometry.

Results are presented for the polygonisation of implicit surfaces defined parametrically and from surface measurements of real objects. Implicit surface representation of real objects is based on the geometric fusion of multiple range images. The Marching Triangles algorithm leads to both representational and computational costs a factor of 3—5 lower than previous *volume based* approaches such as MC.

1 Introduction

1.1 Volume-Based Implicit Surface Polygonisation

Polygonisation of implicitly defined surfaces has attracted considerable interest for visualisation in many fields including computer graphics and medical imaging. Implicit surface polygonisation based on a uniform subdivision of space are widely used [6]. These approaches are *volume-based* as they use a volumetric decomposition of the space to polygonise the implicit surface. A polygonal representation is constructed by intersecting the implicit surface with cubic or tetrahedral cells. Vertices in the polygonal representation are positioned at the intersection of the edges of the volume decomposition with the implicit surface. Each polygon is then subdivided into a set of triangles. The surface to volume intersection produces a highly non-uniform distribution of triangle shape and size, resulting in a highly inefficient representation of the implicit surface geometry. Representation accuracy is dependent on the size of the spatial subdivision. The use of a uniform spatial subdivision limits the maximum polygon size to the size of the volume cell. This results in polygonal representations which are highly inefficient for the geometric accuracy obtained. Typical medical images result in meshes of order 10^6 triangles. Adaptive spatial subdivision and mesh optimisation techniques have been proposed to obtain more efficient representations from volume-based techniques. A comparative evaluation of volume-based algorithms is given in [7].

[1]Supported by EPSRC GR/K04569 'Finite Element Snakes for Depth Data Fusion'

1.2 Surface-Based Implicit Surface Polygonisation

In this paper we present an alternative to previous volume-based implicit surface polygonisation approaches. A surface-based approach called 'Marching Triangles' (MT) is introduced which enables implicit surface polygonisation based on the local surface geometry and topology. Surface-based approaches, as defined by Boissonnat [2], allow mesh vertices to be placed according to the local surface geometry. A 3D Delaunay constraint is introduced which ensures the connectivity between mesh vertices is locally Delaunay and correctly approximates the implicit surface. The resulting triangulation is an accurate and efficient representation of the implicit surface. The triangulation is an approximation of the Delaunay triangulation for the set of mesh vertices on a manifold surface. The Delaunay triangulation is an optimal geometric structure which gives uniform triangle shape, resulting in an efficient representation. The resulting triangulation is therefore directly suitable for applications such as visualisation or finite element analysis.

In this paper we introduce the Marching Triangles (MT) algorithm for implicit surface triangulation with approximately uniform triangle size and shape. Results are compared with the volume-based Marching Cubes (MC) approach to implicit surface polygonisation with uniform spatial decomposition [1]. MT with an adaptive distribution of triangle size according to the local surface curvature will be investigated in future work.

The motivation for this work is the reconstruction of 3D models of real objects from multi-view sets of range image surface measurements. Recent research [5, 3] has resulted in the geometric fusion of multi-view range images into a single implicit surface representation. Reliable geometric fusion in regions of complex geometry is achieved by constraints on the local surface geometry and topology based on measurement uncertainty [5]. In previous work the MC approach was used to reconstruct a triangulated model from the implicit surface representation. This approach has several limitations: the surface is a closed manifold; all data are required a priori; the resulting triangulation is highly non-uniform; the method is computationally expensive and reconstruction of object geometry is limited by the cube size. MT was developed to overcome these limitations.

2 Implicit Representation of an Open Manifold Surface

An implicit surface is defined as the zero-set, $f(\vec{x}) = 0$, of a field function, $f(\vec{x})$, where $\vec{x}$ is a point in 3D space R^3. A closed manifold surface is a 2-manifold in R^3 of arbitrary topology which is geometrically continuous in position. An open manifold surface has geometric discontinuities in position i.e. holes. Implicit representation of a manifold surface, S, can be achieved by defining a scalar field function, $f(\vec{x})$, to be the distance to the nearest point on the surface, $\vec{x}_{nearest}$, where $f(\vec{x}_{nearest}) = 0$: $f(\vec{x}) = |\vec{x}_{nearest} - \vec{x}|$. Signed field function representations are commonly required for implicit surface polygonisation algorithms, such as Marching Cubes, because they define points as either 'inside', $f(\vec{x}) < 0$, or 'outside', $f(\vec{x}) > 0$, a closed manifold surface S. A change in the sign of the field function between two points indicates that the manifold surface must intersect a straight line between those points at least once. A signed field function can be evaluated for a closed manifold surface, S, from the surface normal orientation, $\vec{n}_{nearest}$, at the nearest point, $\vec{x}_{nearest}$ as: $f(\vec{x}) = (\vec{x} - \vec{x}_{nearest}).\vec{n}_{nearest}$. Representation of an open manifold surface requires explicit representation of the discontinuity boundary. A boundary function $b(\vec{x})$ is defined which is '*false*' if the nearest point is internal to the surface and '*true*' if the nearest point is on the boundary. Thus an open manifold surface, S, is represented in an implicit form as the set of points for which $[f(\vec{x}) = 0, b(\vec{x}) = \text{'}false\text{'}]$. Representation of an open manifold surface, S, is illustrated for a cross-section in figure 1. Field function evaluation is shown for two points x_1 and x_2 with nearest points internal to the surface, $b(x_1) = \text{'}false\text{'}$, and on the surface boundary, $b(x_2) = \text{'}true\text{'}$.

In the implicit surface-based geometric fusion approach [5] information on surface boundary, $b(\vec{x})$, and orientation, $\vec{n}_{nearest}$, is available when evaluating the field function $f(\vec{x})$ at no additional computational cost. For parametric implicit surface functions this information is defined implicitly in the continuous field function, $f(\vec{x})$, and may be found by a numerical scheme such as gradient decent. This is analogous to edge intersection techniques used in volume-based implicit surface polygonisation schemes.

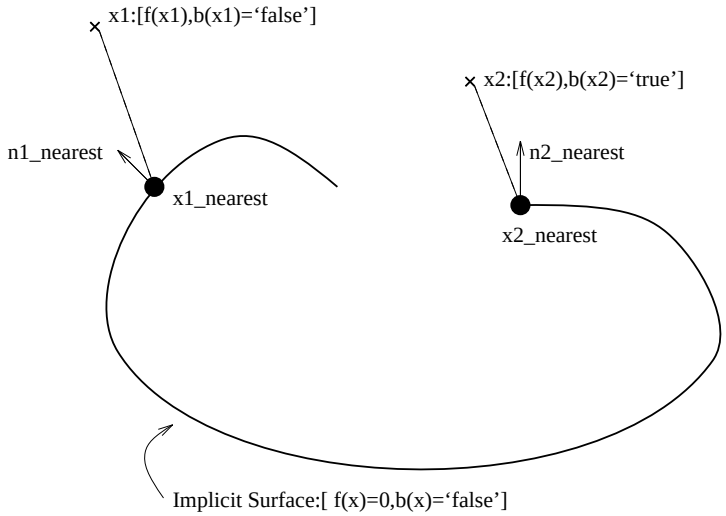


Figure 1: Field function evaluation

3 Delaunay Triangulation on a Manifold Surface

This section defines the theoretical basis of a local 3D procedure for constructing a triangulated polyhedral model, $M = \{T_0 ... T_N\}$, of an unknown object surface, S. We introduce a 3D surface-based constraint for the Delaunay triangulation of a set of points, $X = \{\vec{x}_1 ... \vec{x}_i ... \vec{x}_N\}$, on an arbitrary topology manifold surface, S, where each point $\vec{x}_i$ is a vector, (x, y, z), displacement from the origin in R^3.

The 3D Delaunay triangulation of an arbitrary point set X is composed of tetrahedral volumes, $T_{ijkl} = T(\vec{x}_i, \vec{x}_j, \vec{x}_k, \vec{x}_l)$, such that there exists a sphere which passes through each vertex, $(\vec{x}_i, \vec{x}_j, \vec{x}_k, \vec{x}_l)$, of T_{ijkl} which does not contain any other interior points of X. Each tetrahedra is composed of four triangular faces, $(T_{ijk}, T_{ilj}, T_{lik}, T_{jkl})$. In the case where the points, X, lie on a manifold surface, S, Boissonnat [2] derives the following important property which must be satisfied for the Delaunay triangulation $D(X)$ to contain a polyhedron, $M(X)$, which is a correct approximation of the surface S. Polyhedron, $M(X)$, is a simple manifold composed of a connected subset of the faces of the Delaunay triangulation, $D(X)$. Correct approximation requires that the polyhedron, $M(X)$, respect the relative locations of the points, X,on the surface S. This requires that $M(X)$ is diffeomorphic to a curved polyhedron $M_C(X)$ tightly stretched on the surface S through the points X. Thus each triangle T in $M(X)$ must be diffeomorphic to a curved triangle T_C on $M_C(X)$. Diffeomorphism between two surface patches A and B can be defined as the existence of a projection that relates each point on A and B and respects their relative locations. A triangle T_{ijk} which is locally diffeomorphic to the manifold surface S is a face in the 3D Delaunay triangulation $D(X)$ provided the following property is satisfied:

> **Delaunay Face Property:** Triangle $T(\vec{x}_i, \vec{x}_j, \vec{x}_k)$ is a face in the 3D Delaunay triangulation $D(X)$ provided there exists a circumsphere passing through each triangle vertex, $(\vec{x}_i, \vec{x}_j, \vec{x}_k)$, that does not contain any other interior point of X.

The above property follows from the definition of the Delaunay triangulation. A proof for the Delaunay triangulation in R^2 is given by O'Rouke [8] and extends to R^3. The Delaunay Face Property requires that the set X of points on a manifold surface must be sufficiently dense to satisfy the above constraint. This requires that in regions of high curvature or for thin parts the maximum spacing between measurements is determined by the amount of free space around the surface. The Delaunay face constraint is illustrated in Figure 2 for a polyhedral approximation $M(X)$ of the manifold surface S. The sphere passing through the vertices of triangle, T_{ijk} does not contain any other point in X.

If the above face property is satisfied for all T in a polyhedron $M(X)$ which is diffeomorphic to surface, S, then the Delaunay triangulation $D(X)$ must contain the polyhedron $M(X)$. Consequently if $M(X)$ is in $D(X)$ then it inherits the optimal geometric properties of the Delaunay triangulation. Thus $M(X)$

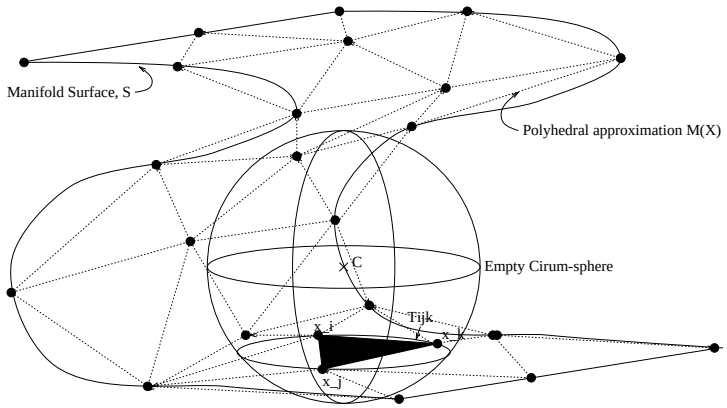


Figure 2: Delaunay face property

is a Delaunay triangulation of the manifold surface, S, for the point set X. The Delaunay triangulation defines an optimal geometric structure which is symmetric, isotropic and closely related to the metric of the surface. Distances on the triangulation $M(X)$ approximate the distance on the manifold surface S. In particular the triangulation maximises the minimum angle of any triangle. This is analogous to the 2D Delaunay triangulation of a point set where the points, X, lie on a 2-manifold surface in R^3 rather than the plane R^2.

4 Delaunay Surface Constraint

Reconstruction of a triangulated polyhedron that correctly approximates the surface and is geometrically optimal requires two conditions to be satisfied. Each triangle T in $M(X)$ must:

1. Respect the relative locations of points X on the surface S.
2. Satisfy the Delaunay Face Property.

In this section we use the above conditions to derive a surface-based approach to triangulation of a manifold implicit surface. We start with a partial model, $M'(X')$, which is a correct approximation of part of the surface S and is also a polyhedron in the Delaunay triangulation $D(X')$, as defined in the previous section. We define a local procedure for adding triangular elements, T_{new}, to the boundary of, M', such that the resulting triangulation $M = \{M', T_{new}\}$ is also a correct approximation of S and is locally Delaunay. Given an edge $e(\vec{x}_i, \vec{x}_j)$ on the boundary of the existing model we find a new point $\vec{x}_{new}$ on the surface region adjacent to e outside the boundary of the existing model. Thus the triangle $T_{new} = T(\vec{x}_i, \vec{x}_j, \vec{x}_{new})$ is a correct local approximation of the surface. For the triangle T_{new} to be added to the model M' the following constraint must be satisfied:

> **3D Delaunay Surface Constraint**: A triangle, $T(\vec{x}_i, \vec{x}_j, \vec{x}_{new})$, may only be added to the mesh boundary, at edge $e(\vec{x}_i, \vec{x}_j)$, if no part of the existing model, M', with the same surface orientation is inside the sphere passing through the triangle vertices, x, about the centre, c_T, where c_T is the circum-centre of the triangle vertices, $(\vec{x}_i, \vec{x}_j, \vec{x}_{new})$, in the plane of the triangle, T. Surface points of the same orientation are defined by a positive normal dot product, $n_T.n_{M'} > 0$.

The Delaunay surface constraint is illustrated in Figure 3. This constraint guarantees that triangle T_{new} uniquely approximates the local surface in the mesh $M(X)$. Thus the mesh $M(X)$ does not over-fold or self-intersect. The resulting triangulation will correctly represent the surface topology and approximate the surface geometry.

The above constraint is an approximation of the Delaunay Face Property. An approximation is required as the Delaunay Face Property does not constrain the centre of the empty circum-sphere. Implementation

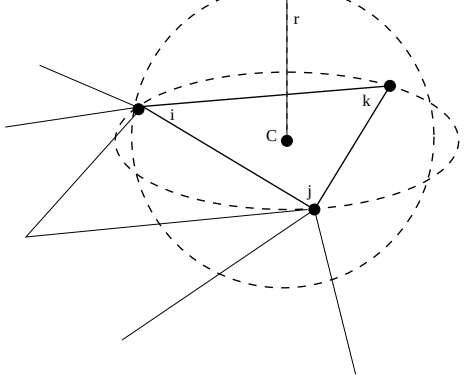


Figure 3: 3D Delaunay Surface Constraint

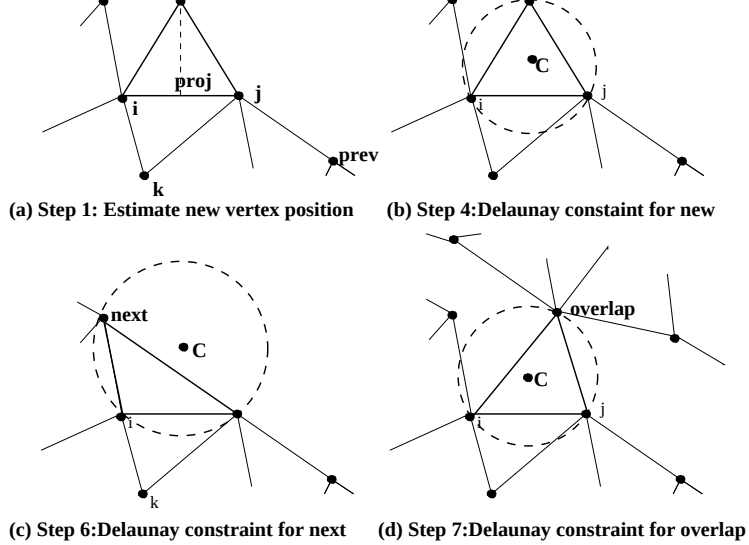
of the exact Delaunay Face Property would require a search for an empty sphere with centre on the line equi-distant from the triangle vertices. Constraining the centre of the circum-sphere to be the centre of the circum-circle in the triangle plane, c_T, enables the Delaunay Face Property to be satisfied. However, this imposes a hard constraint that free-space around the surface must be greater than the radius of the circum-sphere. Relaxation of this constraint is achieved by allowing parts of the existing model, $M'(X')$, with opposite orientation to intersect the sphere. This allows triangulation of complex geometries such as thin surface sections and region of high curvature. The resulting triangulation is locally approximately a Delaunay triangulation of the manifold surface S.

This constraint ensures that the triangle T_{new} is locally a Delaunay triangulation of the manifold surface, S. This ensures that the resulting triangulated model M, is approximately globally Delaunay, Fortune [4]. Thus the resulting model $M(X)$ will approximate the optimal geometric properties of the Delaunay triangulation D(X) for the point set X. In particular the resulting triangulation will have the desirable properties of neighbourhood symmetry and isotropy. This constraint does not impose any restrictions on the mesh vertex position. In particular mesh vertices are not restricted to lie at measurement points unlike previous algorithms [2, 9]. This facilitates the design of a new algorithm which positions mesh vertices according to local surface geometry.

5 Marching Triangles Algorithm

An implicit surface triangulation algorithm can now be developed based on the local 3D Delaunay surface constraint. Given an implicit surface representation, $[f(\vec{x}), b(\vec{x})]$, of an arbitrary topology open manifold surface the triangulation algorithm proceeds as follows. Firstly an initial seed model, $M(X) = M_0(X_0)$, is defined. This may be either a single triangular seed element or a previously constructed model to which we wish to incorporate new measurements. The current model $M(X)$ is represented as a list of edges and vertices. The algorithm is implemented as a single pass through the edge list. New edges introduced by the addition of new elements to the model are appended to the end of the edge list. The algorithm does not terminate until all model edges have been tested once. The algorithm proceeds by testing each edge, $e_{bound} = e(\vec{x}_i, \vec{x}_j)$, on the current model boundary, M:

1. Estimate a new vertex position, $\vec{x}_{proj}$, by projecting a constant distance, l_{proj}, perpendicular to the mid-point of the boundary edge, e_{bound}, in the plane of the model boundary element, $T_{bound} = T(\vec{x}_j, \vec{x}_i, \vec{x}_k)$.
2. Evaluate the new vertex position, $\vec{x}_{new}$, on the implicit surface as the nearest point to $\vec{x}_{proj}$: $\vec{x}_{new} = \vec{x}_{nearest}$ where $f(\vec{x}_{nearest}) = 0$.
3. Terminate the mesh growing (8) for the edge e_{bound} if either:
   (a) Nearest point is on the boundary $b(\vec{x}_{new}) =$'$true$'.
   (b) Implicit surface normal orientation, n_{new}, of $T_{new} = T(\vec{x}_i, \vec{x}_j, \vec{x}_{new})$ is opposite to the model orientation n_{bound} of the boundary triangle T_{bound}: $n_{bound}.n_{new} < 0$.
4. Apply 3D Delaunay Surface Constraint to T_{new}.
5. If T_{new} passes the 3D Delaunay Surface Constraint



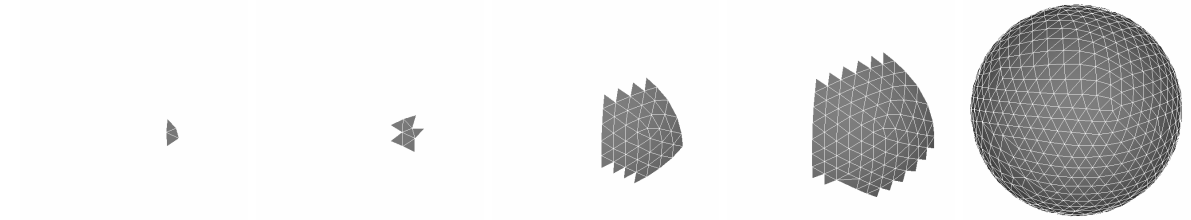
(a) Step 1: Estimate new vertex position (b) Step 4:Delaunay constaint for new

(c) Step 6:Delaunay constraint for next (d) Step 7:Delaunay constraint for overlap

Figure 4: Illustration in 2D of key steps in the Marching Triangles algorithm

   (a) add vertex $\vec{x}_{new}$ to the vertex list.
   (b) add triangle T_{new} to the mesh M.
   (c) add edges $e(\vec{x}_j, \vec{x}_{new})$ and $e(\vec{x}_{new}, \vec{x}_i)$ to the end of the edge list.
6. If T_{new} fails the Local Delaunay Surface Constraint then apply steps 4&5 to adjacent boundary vertices, $T_{new} = T_{prev} = T(\vec{x}_i, \vec{x}_j, \vec{x}_{prev})$ or $T_{new} = T_{next} = T(\vec{x}_i, \vec{x}_j, \vec{x}_{next})$.
7. If T_{new}, T_{next} and T_{prev} all fail the Local Delaunay Surface Constraint then if the Delaunay circumsphere for $T_{new}(\vec{x}_i, \vec{x}_j, \vec{x}_{new})$ overlaps a boundary triangle, $T_{overlap}$, on an existing part of the model, $M(X)$ with the same orientation as the boundary triangle, T_{bound}, such that $n_{bound}.n_{overlap} > 0$, then apply steps 4&5 with $T_{new} = T(\vec{x}_i, \vec{x}_j, \vec{x}_{overlap})$ where $\vec{x}_{overlap}$ is the nearest boundary vertex on $T_{overlap}$.
8. Testing of edge e_{bound} terminates when one or no new triangles have been added to the model, M.

The mesh growing algorithm defined above enables triangulation of a manifold implicit surface of arbitrary topology and geometry. The principal steps of the MT algorithm are illustrated in Figure 4. Steps (1) and (2) estimate a possible new vertex position, x_{new}, on the implicit surface, $f(\vec{x}_{new})$. Projection by a constant distance ensures approximately uniform triangle size. Step (3) ensures that the new triangle, T_{new}, is a correct approximation of the local surface. New mesh vertices must correspond to non-boundary points on the implicit surface, step (3a). This constraint ensures that the the boundaries of geometric discontinuities in an open manifold implicit surface are correctly reconstructed. Step (3b) ensures that the local model geometry corresponds to the implicit surface geometry. This is required where the estimated vertex position, $\vec{x}_{proj}$, may erroneously correspond to a different part of the object surface. This may occur for thin object parts with the simple estimation scheme of step (1). An adaptive scheme for estimation of new vertex positions based on local surface properties such as curvature would reduce incorrect correspondences. Step (4) ensures that the new triangle, T_{new}, locally satisfies the Delaunay circumsphere constraint. Hence, the resulting triangulation is locally Delaunay. Step (5) adds a triangle satisfying the Delaunay constraint to the mesh. New boundary edges are appended to the end of the edge list to ensure that the mesh growing algorithm will test them prior to termination. Step (6) allows the local connection of existing model vertices to form a continuous surface representation. Step (7) is required to eliminate cracks in the implicit surface where the estimated triangles on adjacent boundary regions overlap. This occurs for mesh growing where different parts of the triangulation meet due to the open manifold surface topology. Step (8) terminates testing for edge e_{bound} after one or no triangles have been added. The mesh growing algorithm then applies steps (1 to 8) to the next boundary edge in the edge list. The 'Marching Triangles' mesh growing algorithm reconstructs a triangulation which is a correct approximation of the implicit surface and is locally Delaunay. The resulting triangulated mesh, $M(X)$, approximates the optimal geometric properties of the Delaunay triangulation for the set of vertices X on the surface S.



(a) 5 triangles (b) 10 triangles (c) 50 triangles (d) 100 triangles (e) 1498 triangles

Figure 5: Marching Triangles for a sphere: (a—d) intermediate meshes, (e) final model

Application of the MT algorithm to a parametrically defined implicit surface representation of a sphere is illustrated in Figure 5. The algorithm is initialised as a single triangular seed on the implicit surface. The mesh growing then proceeds to add elements to the mesh boundary. Figures 5(a—d) show snapshots of the mesh growing procedure for 5,10,50 and 100 triangles. The algorithm terminates when no more triangles can be added to the mesh boundary 5(e) 1498 triangles. In this case the implicit surface is closed and terminates when no boundary edges exist.

MT does not impose any constraint on the position of new vertices. In particular mesh vertices are not constrained to lie at measurement points unlike previous mesh growing procedures [2, 9]. This facilitates adaptive mesh growing by evaluating the projected distance, l_{proj}, according to the local surface geometry. Results presented in this paper are for a constant projection distance which enables a uniform distribution of triangle size and shape. In principle an adaptive Marching Triangles algorithm could be implemented where the projection distance is derived from local surface properties such as curvature.

If the implicit surface is composed of several disconnected surface regions then a 'seed' mesh is required for each region. An analogous scheme is required with previous volume-based implicit surface polygonisation algorithms such as MC [1] and mesh growing algorithms [9].

The MT algorithm allows either 'static' or 'dynamic' integration. This is advantageous in situations where the implicit surface is constructed for the geometric fusion of multiple sets of measurements [3, 5]. Dynamic integration allows the addition of new measurements into an existing model. MT enables the extension of an existing model boundary if new measurements modify the implicit surface in an adjacent region. Dynamic integration is possible as the mesh growing algorithm acts directly on the existing triangulated model. Previous volumetric approaches polygonise the implicit surface by the intersection with a volume cell. Dynamic integration is therefore not possible unless the intermediate volumetric data structure is maintained.

6 Algorithm Performance

6.1 Computational Complexity

The computational complexity of implicit surface polygonisation algorithms can be defined in terms of the number of field function evaluations. The time complexity of the MT algorithm depends on the number of edges, N_E in the polyhedral model, M. The addition of new elements to each edge in the model is considered when it is on the boundary of an intermediate partial model. Testing of a candidate triangle to be added to the model boundary requires two implicit surface function evaluations. One to define the nearest point on the surface and the second to determine model overlap. The Euler formula [8] for a trianglated mesh defines the relationship between the number of edges, N_E, nodes, N_N, faces, N_F, and handles, N_H: $N_N - N_E + NF = 2 - 2N_H$. It follows that if the number of handles is small the number of nodes and faces is the same order as the number of edges $O(N_N) = O(N_E) = O(N_F)$. For MT with a constant projection distance, l_{proj}, the element edge length and area are approximately constant. Therefore, the computational complexity is proportional to the area of the manifold surface S. In addition the number of elements in the final representation is proportional to the surface area.

The computational complexity of the volume-based MC approach depends on the number of cubes visited. This depends on the number of cells intersected by the implicit surface. The lower bound on the number of cells occurs for a flat implicit surface aligned with the volumetric subdivision. In this

Object	Model Size		Time(s) (SUN Sparc 10)	
	MT	MC	MT	MC
Sphere	1498	11272	4	12
Torus	1198	8744	4	13
Jack	4533	13032	226	574
Telephone	6178	41759	43	824
Rabbit	9817	26792	106	1180
Teapot	33728	78507	795	2785
Soldier	49922	82877	1087	4191

Table 1: Comparison of Marching Triangles and Cubes

case the lower bound on the number of cells visited is proportional to the area of the surface. For a curved surface the number of cells visited is proportional to a factor times the area of the surface. On average approximately two implicit surface function evaluations are required per cube visited. Thus the computational complexity of the MT algorithm is equal to the lower bound of the complexity for the MC algorithm.

6.2 Representation Cost

The same representation accuracy for surface S is achieved if the constant cube size for MC is equal to the constant projection distance for the MT algorithm. The MC algorithm generates at least two triangles per cube. This defines the lower limit for the number of triangles which is proportional to the implicit surface area. In general for a curved surface the number of triangles for the MC algorithm will be considerably larger than the lower limit [7]. For MT the number of triangles will always be proportional to the area of the surface as the positioning of vertices is based on the local surface geometry. The number of triangles is approximately geometrically optimal for a given manifold surface as it approximates the metric of the surface,S, due to the local Delaunay constraint. Hence the number of triangles for MT is equal to the lower bound of the MC algorithm for the same representation accuracy. In practice the number of triangles is significantly lower for MT due to the uniform distribution of triangle shape.

6.3 Geometric Limitations

The MT algorithm reconstructs the correct topology of implicit surface features larger than the constant projection distance, l_{proj}. Assuming the implicit surface correctly represents the local topology then the lower limit for correct topology reconstruction is the projection distance. The 3D Delaunay Surface Constraint does not impose any limitations on the distance between adjacent surfaces. The use of this constraint enables correct reconstruction of arbitrarily thin object parts and crease edges. The MT algorithm eliminates limitations on surface geometry inherent in previous mesh growing algorithms [2, 9]. The cell size for volume-based algorithms imposes similar limitations on reconstruction of correct topology [7]. Features or thin objects smaller than the cell size may result in incorrect reconstruction and spurious artifacts.

7 Results

Direct comparison of the representational and computational efficiency of the MT and MC has been performed for polygonisation of implicit surfaces derived from synthetic models and real object measurements. Throughout this comparison the MT projection distance, l_{proj}, is equal to the MC voxel size to obtain the same geometric representation accuracy.

7.1 Parametric Implicit Surfaces

Results for three parametric implicit surfaces derived from algebraic expressions for a sphere, torus and jack are given in Table 1. Figure 6 gives a comparison of the reconstructed triangulated mesh using the

MT and MC approaches. Representation costs for MT are a factor of 3—7 lower for all shapes. The sphere and torus gives the greatest difference in representation cost between MT and MC due to the smoothly curved surfaces. The computational cost for MT is reduced by a factor of 3 compared to MC. This is due to the use of a surface-based approach where the cost is directly proportional to the surface area. Comparison of the reconstructed meshes illustrates the improvement in representation efficiency due to uniform triangle shape achieved with the MT approach. Results demonstrate that both the representation and computational cost are significantly lower for the MT approach for the same representation accuracy and visulisation quality.

7.2 Implicit Surfaces from Geometric Fusion

Results for the MT implicit surfaces polygonisation of surface measurements of real objects are illustrated in Figure 7. Implicit surface representations were obtained by geometric fusion of multiple range images [5]. The telephone and bunny data sets were measured using a Cyberware range sensor [3] and the teapot and soldier data sets are from an NRCC range sensor [10]. Each data sets contains approximately 10 range images. The teapot and soldier data sets are taken from multiple viewpoints in the horizontal plane, resulting in regions of no data for horizontal surfaces and occluded regions.The MT projection distance and MC cube size are equal to the sampling resolution of the range images. Results demonstrate accurate reconstruction of both surface geometry and topology. Small holes, thin surface regions, crease edges and regions of high curvature are correctly reconstructed. The relative representation and computation costs for MC and MT at the same resolution are given in Table 1. The representational costs are a factor of 2—7 lower for MT compared to MC. Objects with smooth surfaces result in a greater reduction in representation cost for the same geometric accuracy. The computational cost for MT is a factor of 3—10 lower than MC as it is directly proportional to the surface area. A comparison of the MT and MC reconstructed mesh for a region of the bunny head is presented in Figure 8. This illustrates that the MT representation efficiency is significantly better due to the uniform triangle shape.

8 Conclusions

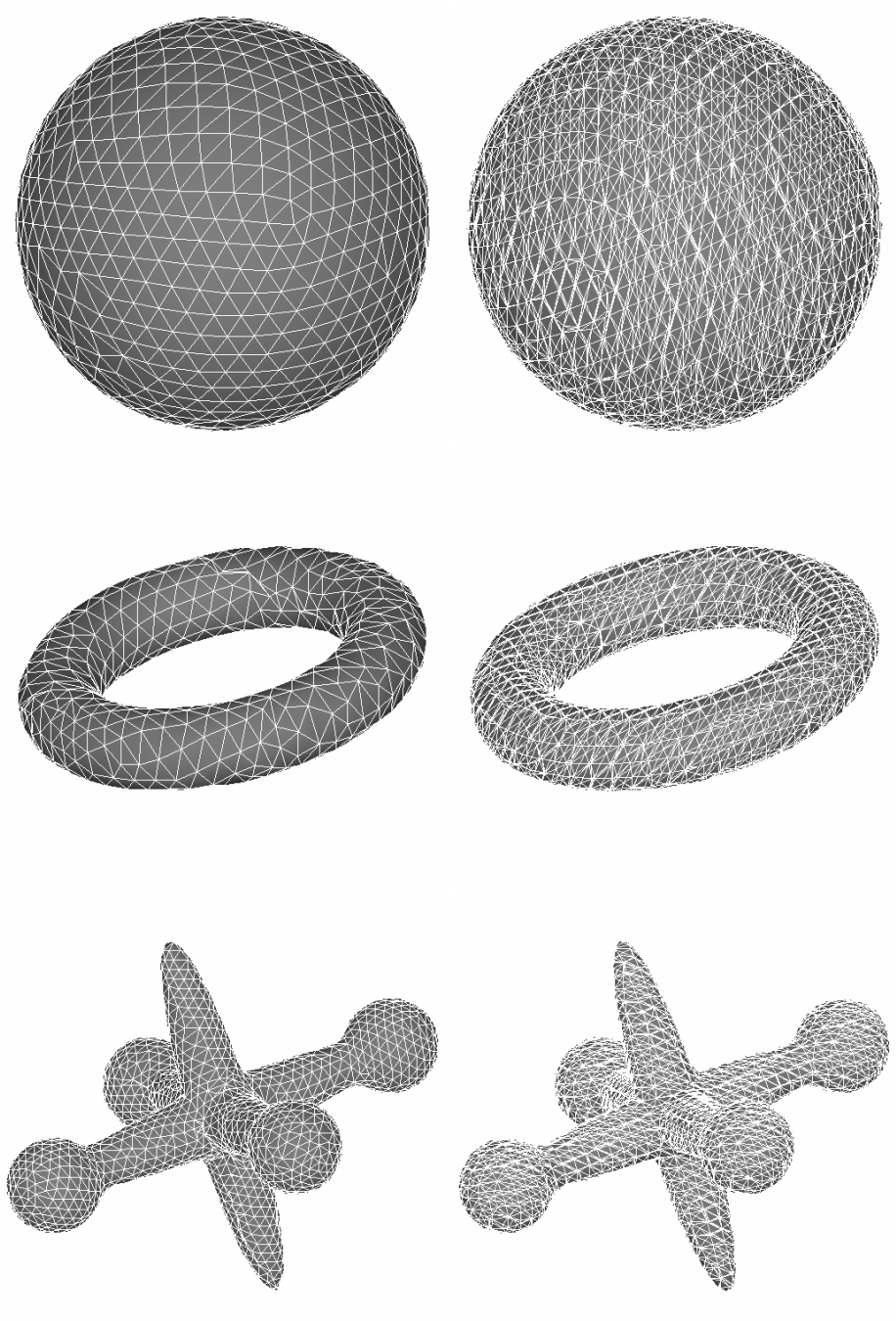
A new 'surface-based' approach to implicit surface polgonisation has been presented called 'Marching Triangles' MT. A local '3D Delaunay Surface Constraint' is used to control mesh growing across a manifold surface in R^3. The manifold surface is open or closed and of arbitrary topology. The MT algorithm ensures that the reconstructed triangulated mesh:

1. Is a correct approximation of the surface geometry and topology.
2. Is locally a Delaunay triangulation of the manifold surface.

The use of a constraint based on the Delaunay triangulation results in a representation with optimal geometric properties. In particular efficient representation is achieved due to uniform triangle shape.

Comparison with 'volume-based' implicit surface polygonisation algorithms such as Marching Cubes demonstrates significant improvement in both representation and computation costs for the same accuracy. Theoretical analysis shows that both costs are equal to the lower bound of the Marching Cubes algorithms. Geometric limitations on reconstruction of the correct surface topology are similar to those with previous volume-based approaches. Results for polygonisation of implicit surfaces defined parametrically and from surface measurements of real objects give a factor of 3—5 improvement in both representation and computation costs.

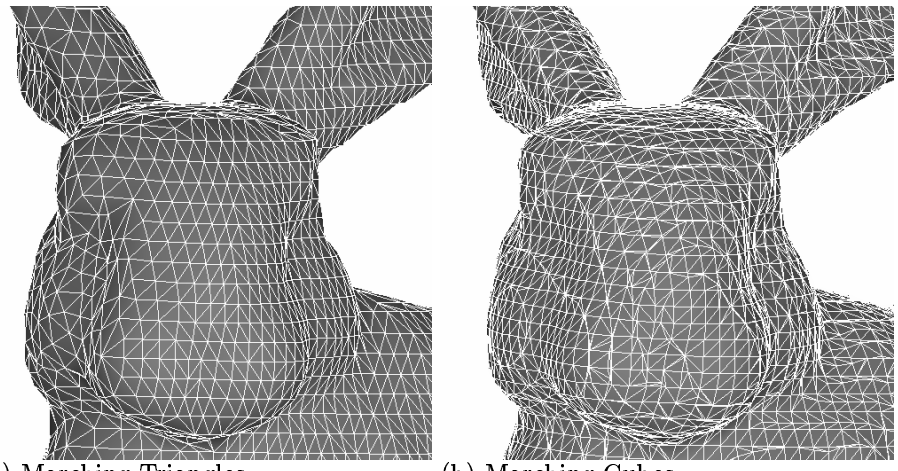
The MT approach also has several potential advantages over previous volume-based approaches. Firstly it allows accurate polygonisation of open manifold surfaces through the use of an implicit boundary function. Secondly dynamic integration of new implicit surface regions into an existing model is achieved. This is important for applications such as fusion of multiple range images of real objects as it allows continuous updating of the model based on new measurements. The MT approach does not impose any restrictions on the mesh vertex position unlike previous mesh growing algorithms [2, 9]. This allows efficient representation with approximately uniform element shape. Future work will investigate adaptive implicit surface polygonisation based on local geometric properties such as surface curvature.



(a) Marching Triangles (b) Marching Cubes

Figure 6: Parametric Implicit Surface Polygonisation

Figure 7: Marching Triangle Reconstruction of 3D Models for Real Objects



(a) Marching Triangles (b) Marching Cubes

Figure 8: Real Object Implicit Surface Polygonisation

References

[1] J. Bloomenthal. An implicit surface polygonizer. *Graphics Gems ed. Heckbert,P.S.*, 4:324—350, 1994.

[2] J.D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266—286, 1984.

[3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Computer Graphics Proceedings, SIGGRAPH*, 1996.

[4] S. Fortune. Voronoi diagrams and deluanay triangulations. In *Computing in Euclidean Geometry, eds. Du, D.-Z. and Hwang, F.*, pages 193—230, 1992.

[5] A. Hilton, A.J. Stoddart, J. Illingworth, and T. Windeatt. Reliable surface reconstruction from multiple range images. In *4th European Conference on Computer Vision*, pages 117—126. Springer, 1996.

[6] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163—169, 1987.

[7] P. Ning and J. Bloomenthal. An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications*, 13(November):33—41, 1993.

[8] J. O'Rouke. *Computational Geometry in C.* Cambridge University Press, 1994.

[9] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 573—580, 1994.

[10] M. Soucy and D. Laurendeau. A dynamic integration algorithm to model surfaces from multiple range images. *Machine Vision and Applications*, 8:53–62, 1995.