MARCHING TRIANGLES: RANGE IMAGE FUSION FOR COMPLEX OBJECT MODELLING

A. Hilton[1], *A.J.Stoddart, J. Illingworth and T.Windeatt*

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 implicit surface polygonisation is introduced in this paper. This is applied to the reconstruction of 3D surface models of complex objects from multiple range images. Geometric fusion of multiple range images into an implicit surface representation was presented in previous work. This paper introduces an efficient algorithm to reconstruct a triangulated model of a manifold implicit surface. A local 3D constraint is derived which defines the Delaunay surface triangulation of a set of points on a manifold surface in 3D space. The 'Marching Triangles' algorithm uses the local 3D constraint to reconstruct a Delaunay triangulation of an arbitrary topology manifold surface. Computational and representational costs are both a factor of 3–5 lower than previous volumetric approaches such as marching cubes.

1. INTRODUCTION

Automatic reconstruction of integrated 3D models from multiple 2.5D range images is a primary goal of recent research [4, 7, 8, 9]. A volumetric approach to geometric fusion based on an intermediate implicit surface representation was recently introduced [3]. This provides a framework for reliable integration of multiple images to ensure consistent representation of the local surface geometry and topology based on measurement uncertainty. In previous work the Marching Cubes approach was used to reconstruct a triangulated model from the implicit surface representation. This approach has four principal limitations: the surface is closed; all data are required a priori; the resulting triangulation is highly non-uniform and the method is computationally expensive. The 'Marching Triangles' surface based implicit surface polygonisation algorithm is introduced in this paper to overcome these limitations. A local 3D constraint is derived which defines the Delaunay triangulation of a set of measurements on a manifold surface in 3D space. A procedure is introduced for growing a triangulation across an implicit surface based on the local 3D constraint. Marching Triangles enables: polygonisation of open surfaces; dynamic integration of new data; efficient representation; reduced computational cost and correct reconstruction of complex surface geometry in regions of high curvature and thin sections.

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

2. THEORETICAL BACKGROUND

This section defines the theoretical basis of a local 3D procedure for constructing a triangulated model, M, of an unknown object surface, S. We define a 3D surface based constraint for the Delaunay triangulation of a set of measurements, X, on an arbitrary topology manifold surface, S.

The 3D Delaunay triangulation of an arbitrary point set $X = \{x_1 ... x_i ... x_N\}$ is composed of tetrahedral volumes, $T(x_i, x_j, x_k, x_l)$, such that there exists a sphere which passes through each vertex, x, of T which does not contain any other interior points of X. In the case where the points, X, lie on a manifold surface, S, Boissonnat [1] derives the following important property. The manifold surface triangulation represented in the 3D Delaunay triangulation is defined by the condition that it is composed of triangles $T(x_i, x_j, x_k)$ such that there exists a circumsphere passing through each vertex, x, that does not contain any other interior point of X. This result defines a 3D triangulated manifold surface which correctly represents the measured surface geometry and topology. The manifold surface triangulation defines an optimal geometric structure which is symmetric, isotropic and closely related to the metric of the surface. In particular the triangulation maximises the minimum angle of any triangle. This is the 3D analogue of the 2D Delaunay triangulation of a point set where the points, X, lie on a manifold surface in R^3 rather than R^2.

The above definition of the manifold surface approximation can be used as the basis for deriving a 'surface-based' approach to model reconstruction. Given a partial model, M', of the surface we define a local procedure for adding triangular elements to the model boundary. We define a local constraint based on the properties of the manifold surface triangulation represented in the 3D Delaunay triangulation:

> **3D Delaunay Surface Constraint**: A triangle, $T(x_i, x_j, x_{new})$, may only be added to the mesh boundary, at edge $e(x_i, x_j)$, 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 center, c_T. Where c_T is the circumcenter of the triangle vertices, (x_i, x_j, x_{new}), in the plane of the triangle, T. Surface points of the same orientation are defined by a positive normal dot product, $n_T.n_{M'} > 0$.

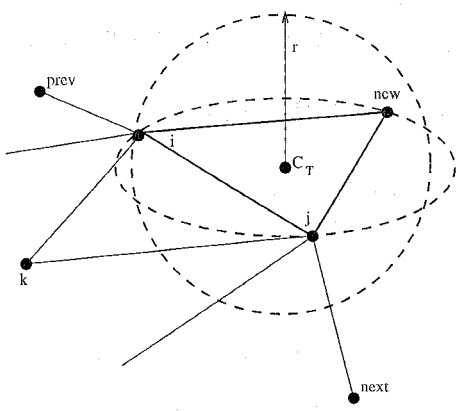


Figure 1: 3D Delaunay Surface Constraint

This constraint is illustrated in Figure 1. This constraint is sufficient to guarantee that each triangle, T, in the resulting model, M, uniquely defines the local surface. Hence, building a model using this constraint guarantees that the local surface does not over-fold or self-intersect. In addition by imposing a constraint that ensures the surface is locally Delaunay we ensure that the resulting triangulated model, M, is globally Delaunay, Fortune [2].

The circumcenter in the triangle plane, c_T, is taken as the fourth point required to define a sphere for each triangular element. A local 3D Delaunay manifold surface is guaranteed provided the triangle, T, satisfies the condition that no part of the existing model is inside an arbitrary sphere passing through the three triangle vertices, [1]. Defining the sphere about the equi-distant point in the element plane enables this constraint to be satisfied. If different surface regions are in close proximity the sphere about the center, c_T, may intersect another surface region but will still give a valid local triangulation. The condition that parts of the existing model, M', with opposite orientation may intersect the sphere allows complex geometries to be modelled.

3. MARCHING TRIANGLES ALGORITHM

An implicit surface triangulation algorithm can now be developed based on the local 3D constraint defined in the previous section. An implicit surface is represented as the zero-set, $f(x) = 0$, of a field function, $f(x)$, which defines the signed distance to the nearest point on the surface for any point in 3D space x. Representation of open manifold surfaces requires explicit representation of the boundary. A boundary function $b(x)$ is defined which is '*false*' if the nearest point is internal to the surface and '*true*' if the nearest point is on the boundary. Integration of multiple range images into a single implicit surface representation was introduced in [3]. Given an implicit surface representation, $[f(x), b(x)]$, of an arbitrary topology manifold surface the triangulation algorithm proceeds as follows. Firstly an initial seed model, $M = M_0$, 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 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, $e_{bound} = e(x_i, x_j)$, on the current model boundary, M:

1. Estimate a new vertex position, x_{proj}, by projection perpendicular to the mid-point of the boundary edge in the plane of the model boundary element $T_{bound}(x_j, x_i, x_k)$, by a constant distance, l_{proj}.
2. Evaluate nearest point on the implicit surface to x_{proj}: $x_{new} = x_{nearest}$ where $f(x_{nearest}) = 0$.
3. Terminate the mesh growing (7) for the edge if:
   (a) Nearest point is on the boundary $b(x_{new}) =$'*true*'.
   (b) Implicit surface orientation at x_{new}, n_{new}, is opposite to the model orientation n_T of $T(x_i, x_j, x_{new})$: $n_T.n_{new} < 0$.
4. Apply 3D Delaunay Surface Constraint to $T_{new} = T(x_i, x_j, x_{new})$.
5. If T_{new} passes the 3D Delaunay Surface Constraint
   (a) add x_{new} to M.
   (b) add T_{new} to M.
   (c) add new edges $e(x_j, x_{new})$ & $e(x_{new}, x_i)$.
6. If T_{new} fails the 3D Delaunay Surface Constraint apply steps 4 &5 to adjacent boundary vertices, $T_{new} = T(x_i, x_j, x_{prev})$ or $T_{new} = T(x_i, x_j, x_{next})$.
7. Testing of e_{bound} terminates when one or no new elements, T_{new}, have been added to the model, M.

The mesh growing algorithm defined above enables triangulation of manifold implicit surface of arbitrary topology and geometry. New mesh vertices must correspond to non-boundary points on the implicit surface, step (3a). This constraint ensures that the measured surface topology is correctly reconstructed. Step (3b) ensures that the local model geometry corresponds to the implicit surface geometry. This is required where the estimated vertex position, 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). Step (5) allows the local connection of existing model vertices to form a continuous surface representation. Marching triangles 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 [1, 7]. This facilitates adaptive mesh growing by evaluating the projected distance, l_{proj}, according to the local surface geometry.

4. COMPLEXITY AND LIMITATIONS

Defining a general form for the computational complexity of any range image integration algorithm is not possible as it is a function of the particular image set, [8, 3]. The worst case computational complexity is approximated for integrating m images of N points assuming $O(N)$ redundant measurements between each pair of images. **Static integration** of

Object	Model Size		Time(s)	
	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

m range images requires $O(m)$ nearest point evaluations for each implicit surface function evaluation, $f(x)$. Assuming the model is triangulated at approximately the same spatial resolution as the original measurements and using the Euler formula [5] the number of implicit surface function evaluations is $O(mN)$. Hence, the time complexity of the static Marching Triangles integration algorithm is $O(m^2N)$.
Dynamic integration of a single range image requires the evaluation of the implicit surface for a single mesh. Assuming no redundancy between the model and range image the worst case complexity for a single image is $O(N)$. The time complexity of dynamic Marching Triangles integration of m range images of N points is $O(mN)$.

The Marching Triangles algorithm reconstructs the correct topology of implicit surface features greater 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 Marching Triangles algorithm eliminates limitations on surface geometry inherent in previous integration algorithms.

5. RESULTS

Direct comparison of the representational and computational efficiency of the Marching Triangles and Marching Cubes implicit surface polygonisation algorithms has been performed for both synthetic and real data. Throughout this comparison the marching triangles projection distance, l_{proj}, is equal to the marching cubes voxel size. Results for three simple implicit surfaces derived from algebraic expressions for a sphere, torus and jack are given in Table 1. Results illustrate that both the representation and computational cost are significantly lower for the Marching Triangles approach.

Results of implicit surface based fusion of multiple range images using Marching Triangles approach are illustrated in Figure 2. Each data sets contains approximately 10 range images. Integration times correspond to a SUN Sparc 20 workstation. The telephone [1] and bunny[1] images were previously used to demonstrate the mesh zippering integration

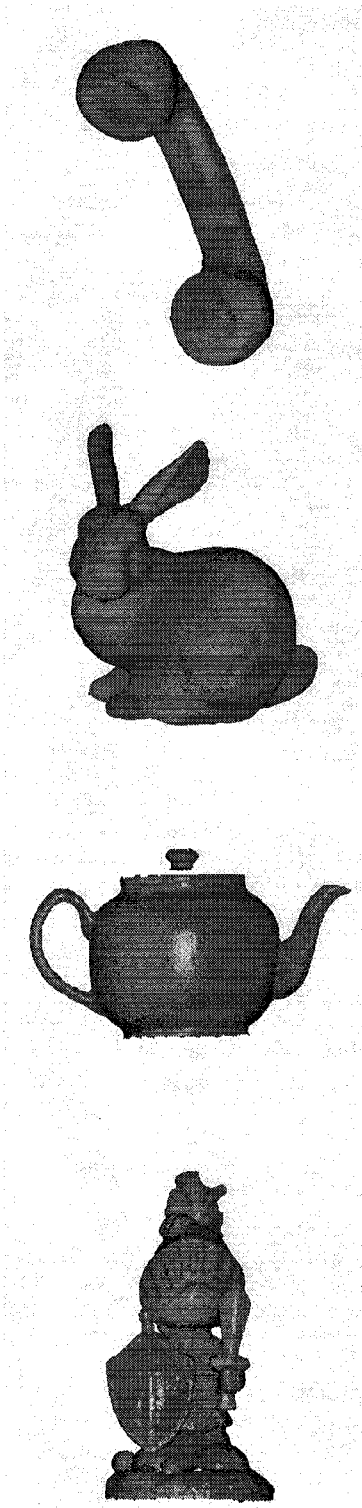
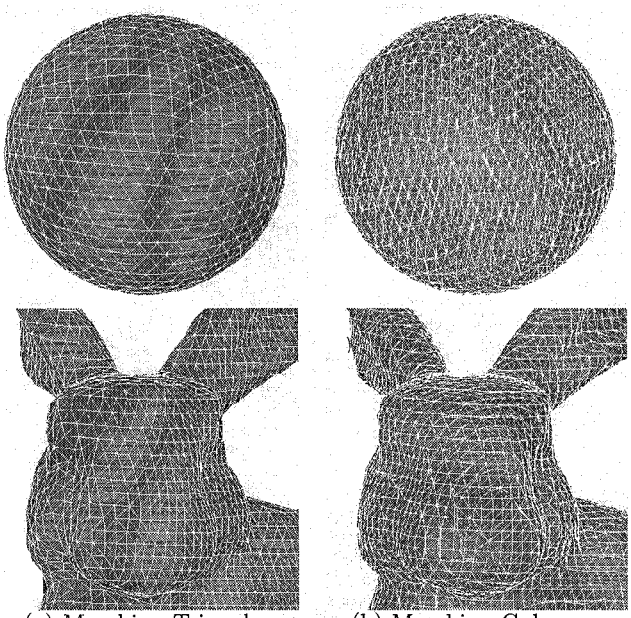


Figure 2: Marching Triangle Reconstruction of 3D Models

[1] Cyberware range scanner data [9]



(a) Marching Triangles (b) Marching Cubes

Figure 3: Comparison of Representation Efficiency

algorithm, [9]. The teapot[2] and soldier[2] data sets were previously used to demonstrate the integration algorithm of Soucy et al. [8]. The teapot and soldier data sets are taken from multiple viewpoints in the horizontal plane, resulting in regions of no data for horizontal surface and occluded regions. Results demonstrate the reliable reconstruction of both surface geometry and topology using the implicit surface based fusion algorithm. Thin surface regions, crease edges and regions of high curvature are correctly reconstructed. Figure 3 provides a comparison of the triangulation obtained with the volumetric and surface based approaches. Statistics of the model reconstruction are summarised in Table 1. The results demonstrate that the use of the Marching Triangles algorithm results in significant reduction in both computation and representation costs.

6. CONCLUSIONS

In this paper we have presented a new method for model reconstruction from multiple range images. Integration consists of two stages. Geometric fusion of multiple range images into a single implicit surface representation which defines the local surface topology and geometry, as presented in previous work [3]. Model reconstruction is then achieved by implicit surface polygonisation. A surface based approach for implicit surface polygonisation has been introduced in this paper and is refered to as 'Marching Triangles'. Previous implicit surface polygonisation algorithms such as marching cubes use volumetric primitives (cubes or tetrahedra). This results in a non-uniform distribution of triangle shape independent of the surface geometry. The Marching Triangles algorithm uses triangular surface primitives resulting in an efficient uniform distribution of triangle shape based on surface geometry. The algorithm is based on a 3D Delaunay Surface Constraint which defines the Delaunay triangulation of a manifold surface in 3D space. This constraint ensures correct reconstruction of the local surface geometry. The resulting triangulation is locally and therefore globally Delaunay. Triangulation is performed by operations in 3D space only. This eliminates constraints on local surface geometry associated with previous mesh growing algorithms which require local 2D projection [4, 7, 8, 9]. Unlike previous integration algorithms model vertices are not constrained to lie at the original measurement positions. Previous volume based implicit surface polygonisation algorithms, such as marching cubes, result in a relatively inefficient representation due to non-uniform triangle shape. Dynamic integration of new measurements into an existing model is also enabled which was not possible with previous polygonisation algorithms. Results demonstrate the reliable reconstruction of surface models of complex objects. Marching Triangles reduces both the computational and representational cost of implicit surface polygonisation by a factor of 3—5 compared to previous approaches. Future work should address the use of Marching Triangles for adaptive implicit surface polygonisation according to the local surface geometry.

[2]NRCC laser range images [6] registered using InnovMetric software [8]

7. REFERENCES

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

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

[3] 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.

[4] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganised points. *Computer Graphics*, 26(2):71—77, 1992.

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

[6] M. Rioux. Laser range finder based on synchronized scanners. *Applied Optics*, 23(21):3837—3844, 1984.

[7] 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.

[8] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(4):344–358, 1995.

[9] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Computer Graphics Proceedings, SIGGRAPH*, 1994.