

Reliable Surface Reconstruction from Multiple Range Images ^{*}

A. Hilton, A.J.Stoddart, J. Illingworth and T.Windeatt

VSSP Group, Department of Electronic and Electrical Engineering
University of Surrey, Guildford. GU2 5XH. U.K.
a.hilton@ee.surrey.ac.uk

Abstract. This paper addresses the problem of reconstructing an integrated 3D model from multiple 2.5D range images. A novel integration algorithm is presented based on a continuous implicit surface representation. This is the first reconstruction algorithm to use operations in 3D space only. The algorithm is guaranteed to reconstruct the correct topology of surface features larger than the range image sampling resolution. Reconstruction of triangulated models from multi-image data sets is demonstrated for complex objects. Performance characterization of existing range image integration algorithms is addressed in the second part of this paper. This comparison defines the relative computational complexity and geometric limitations of existing integration algorithms.

1 Introduction

Recent research has resulted in the independent publication of several algorithms that reconstruct triangulated 3D surface models of complex objects [4, 6, 7, 9]. The goal of surface reconstruction is to estimate a manifold surface, S' , that approximates an unknown object surface, S , using a sample of points, $\mathbf{x} = (x, y, z)$, in 3D Euclidean space, $X = \{\mathbf{x}_0, \dots, \mathbf{x}_{N-1}\}$, combined with knowledge about the sampling resolution, $\Delta\mathbf{x}$, measurement error, ε , and measurement confidence, $p(S'/\mathbf{x}_i)$. Given a method we want to be able to specify conditions on the original surface, S , and sample, X , that allow S' to be a reliable model.

Hoppe et al. [4] presented a general method for constructing an implicit surface representation from unstructured 3D points. Polygonal models were then generated using a ‘marching cubes’ approach [1]. The algorithm is ‘static’ in the sense that all the image data is required prior to the polygonisation process. Soucy et al. [7] integrated range images using canonic subsets of the Venn diagram. The canonic subsets each represent the overlap between a subset of the 2.5D range images and are associated with a 2D viewpoint reference frame. The 2D reference frames are used to eliminate redundant data and merge intersecting regions. Soucy et al. [8] extended this algorithm to be ‘dynamic’ allowing the sequential integration of new range images. Turk et al. [9] integrated range image triangulations using a dynamic mesh ‘zippering’ approach. Overlapping regions of meshes are eroded and the boundary correspondence found by operations in

^{*} Supported by EPSRC GR/K04569. ‘Finite Element Snakes for Depth Data Fusion’

3D space. A local 2D constrained triangulation is then used to join overlapping mesh boundaries to form a single mesh. Boissonnat [2] and Rutishauser et al. [6] retriangulate two overlapping meshes using local 2D constraints on triangle shape. The four integration algorithms are completely different in their approach to constructing a single triangulated model. This results in different complexity, limitations and failure modes.

A new range image integration algorithm is presented in section 2. This is based on a continuous implicit surface representation which combines geometry and topology information from individual range images. This is the first integration algorithm which uses operations entirely in 3D space. Unlike all previous approaches the method does not require either local or global projection to 2D sub-planes. This eliminates limitations on local surface geometry inherent in previous approaches. Performance characterization of existing integration methods is presented in section 3. This comparison defines the relative computational complexity and geometric limitations for reliable reconstruction.

2 New Integration Algorithm

The new range image integration algorithm based on a ‘continuous implicit surface’ is presented in section 2.1. A triangulated model is constructed using a standard implicit surface polygonisation algorithm, section 2.2. Results for the reconstruction of complex objects are presented in section 2.3. The computational complexity and limitations of this approach are discussed in section 3.

2.1 Continuous Implicit Surface Construction

An implicit surface is defined as the zero-set of a scalar field function $f(\mathbf{x}) = 0$. The aim of representing a set of surface measurements, $X = \{\mathbf{x}_0, \dots, \mathbf{x}_{N-1}\}$, as an implicit surface is to construct a smooth field function, $f(\mathbf{x})$, such that the zero-set approximates the data X as closely as possible. A piecewise continuous implicit surface function for multiple range images is presented in this section. This is based on the integration of multiple overlapping 2.5D triangulations. Accurate representation of surface geometry is achieved by combining overlapping measurements according to their confidence. Boundary information is integrated from multiple meshes to obtain an explicit representation of the measured surface topology.

Meshes, M_i , are initially constructed from each range image using a step discontinuity constrained triangulation, [6, 7, 9]. Range images are triangulated in the 2D image plane using a constant distance threshold, $t_d = n\Delta\mathbf{x}$. This defines the local surface continuity for each range image.

For a single mesh M the field function, $f(\mathbf{x})$, is constructed as the signed distance to the nearest mesh point, $\mathbf{p}(x, y, z)$. A binary function, $b(\mathbf{x}) = [0, 1]$, is used to explicitly label field function values, $f(\mathbf{x})$, with nearest points on the mesh boundary. If the nearest point, \mathbf{p} , is not on the mesh boundary, $b(\mathbf{x}) = 0$, the signed distance is the dot product of the vector to the nearest point, $(\mathbf{x} - \mathbf{p})$,

with the surface normal, \mathbf{n}_p , at the nearest point: $f(\mathbf{x}) = (\mathbf{x} - \mathbf{p}) \cdot \mathbf{n}_p$. Alternatively, if the nearest point, \mathbf{p}_b , is on the mesh boundary, $b(\mathbf{x}) = 1$, the signed distance function is evaluated as the sign of the dot product of the vector to the nearest points, $(\mathbf{x} - \mathbf{p}_b)$, with the nearest point normal, \mathbf{n}_{p_b} , multiplied by the Euclidean distance: $f(\mathbf{x}) = \text{sign}[(\mathbf{x} - \mathbf{p}_b) \cdot \mathbf{n}_{p_b}] \times |(\mathbf{x} - \mathbf{p}_b)|$. The zero-set of the field function, $f(\mathbf{x})$, for a single mesh, M , is thus a piecewise continuous function with the same topology as the mesh. The implicit surface representation for a single mesh is illustrated in Figure 1(a) for a 2D cross section.

Integration of multiple range images requires the construction of an implicit surface function based on multiple overlapping meshes M_k where $k = 0 \dots M - 1$. A field function $f_k(\mathbf{x})$ can be implemented as described above for a single mesh. The problem is then to integrate the individual field functions into a single continuous surface, $f(\mathbf{x})$, and boundary label, $b(\mathbf{x})$. This is achieved by first evaluating $f_k(\mathbf{x})$ for each individual mesh and then integrating them using a simple set of rules based on local surface geometry:

- i) Evaluate the signed field function, $f_k(\mathbf{x})$, and boundary function, $b_k(\mathbf{x})$, for each mesh M_k for $k = 0 \dots m - 1$.
- ii) Find the nearest non-boundary mesh point, $b_k(\mathbf{x}) = 0$, from the set $f_k(\mathbf{x})$, $f_{min}(\mathbf{x})$.
- iii) If a non-boundary point, $f_{min}(\mathbf{x})$, does not exist return the nearest boundary point, $b_k(\mathbf{x}) = 1$: $f(\mathbf{x}) = f_{min_{boundary}}(\mathbf{x})$ and $b(\mathbf{x}) = 1$.
- iv) Else find all non-boundary points, $b_k(\mathbf{x}) = 0$, with the same orientation as the nearest point, $f_{min}(\mathbf{x})$: $F = \{f_{same_i}(\mathbf{x})\}$ $i = 0 \dots N_{same}$, where $n_{min}.n_{same_i} > 0$
- v) Find the nearest non-boundary point with opposite orientation, $f_{opposite}(\mathbf{x})$, where $n_{min}.n_{opposite} < 0$.
- vi) Eliminate all points in F , where $f_{same_i}(\mathbf{x}) > f_{opposite}(\mathbf{x})$.
- vii) Evaluate the nearest point as a weighted average of all points in F : $f(\mathbf{x}) = \sum_k w_k f_k(\mathbf{x})$ where $\sum w_k = 1$ and $b(\mathbf{x}) = 0$

This set of rules enables the integration of overlapping meshes to define a continuous zero-set of the field function, $f(\mathbf{x}) = 0$, in all regions where a mesh is continuous and non-zero elsewhere. The rules account for the special cases of two overlapping surfaces with different orientations and multiple overlapping surfaces. The field function evaluation according to the rules given above is illustrated schematically in Figure 1 for the different cases of overlapping surface. Step (iii) explicitly defines mesh boundaries, $b(\mathbf{x}) = 1$, providing an integrated representation of the local surface topology. Steps (iv—vi) eliminate ambiguity if there are multiple overlapping meshes corresponding to different surface regions. This enables correct representation for surface regions of high curvature and different surfaces in close proximity, section 3.2. The weighted average of nearest points, step (vii), enables smooth integration of overlapping meshes using estimates of measurement confidence or blending functions, [3].

2.2 Implicit Surface Polygonisation

Polygonisation of implicit surfaces has received considerable interest for visualisation in medical imaging and computer graphics. The ‘Marching Cubes’ algorithm

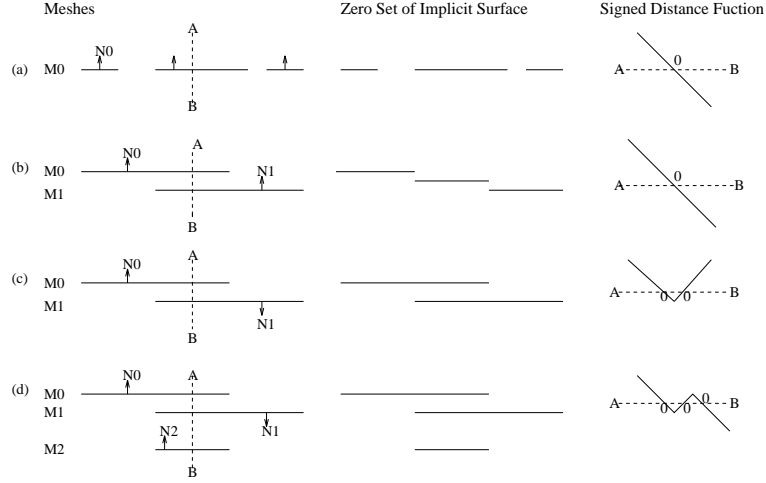


Fig. 1. Continuous Implicit Surface Function: (a) Single mesh, (b) Two meshes from same surface, (c) Two meshes from different adjacent surface with opposite orientation, (d) Three meshes from three different adjacent surfaces.

uses a uniform subdivision of 3D space to reconstruct a triangulated model of a manifold surface without boundaries, [1]. Extension of marching cubes to the polygonisation of a bounded implicit surface was addressed by Hoppe et al. [4]. The modified marching cubes algorithm only reconstructs the implicit surface for cubes which do not intersect the implicit surface boundary. Application of a modified marching cubes algorithm to the mesh based implicit surface enables a 3D triangulated model to be constructed. Explicit representation of the surface boundary, $b(\mathbf{x}) = 1$, together with the implicit surface, $f(\mathbf{x}) = 0$, enables guaranteed reconstruction of a polygonal model with the same topology as the surface measurements. To ensure correct reconstruction of surface features the spatial subdivision used in the marching cubes algorithm must be less than the sampling resolution, $\Delta\mathbf{x}$. Previous implicit surface based approaches [4] did not use a continuous representation, resulting in reconstruction errors, section 3.2.

2.3 Results

Results of the reconstruction process for four objects are shown in Figure 2. All models were constructed from 8—10 range images. The bunny² and telephone² model consist of approximately 25000 triangular elements. This data was previously used to demonstrate the mesh zippering algorithm, [9]. The teapot³ and soldier³ consist of 15000 and 80000 triangular elements. This data was previously used to demonstrate the Canonic subsets algorithm, [8]. The results

² Cyberware scanner range data registered using ICP [9]

³ NRCC scanner range data [5] registered using InnovMetric software [8]

demonstrate that the integration algorithm correctly reconstructs the surface topology for features greater than the sampling resolution, $\Delta\mathbf{x}$. Holes in the original data are correctly preserved in the reconstructed model. Surface regions of high curvature are correctly reconstructed as continuous surfaces. Different surfaces in close proximity, that occur for thin object regions, are also correctly reconstructed. This overcomes the limitations of previous integration methods, [4, 6, 9], in reconstructing surfaces of complex geometry, section 3.3. These results indicate that the integration algorithm reliably reconstructs the underlying surface topology. The computation time for the reconstruction of an integrated model of 25000 elements on a Sun sparc 10 was approximately 12 minutes. This is comparable to previously reported integration times [8].

3 Performance Characterization

A comparative analysis of integration algorithms is presented in this section. Hilton [3] defines integration algorithms in a common framework for direct comparison. Implementation is also discussed including requirements that were previously undocumented. The comparison of integration algorithms presented here focuses on two principal issues. Firstly, the computational complexity of each of the algorithms, section 3.1. Secondly, identification of inherent limitations in each of the integration methods, section 3.2. The comparative analysis considers the following algorithms:

- I: Point-Normal Implicit Surface (Hoppe et al. [4]).
- II: Mesh Implicit Surface (this paper).
- III: Canonic Views (Soucy et al. [7]).
- IV: Mesh Zippering (Turk et al. [9]).
- V: Mesh Growing (Rutishauser et al. [6]).

3.1 Time Complexity

Defining a general form for the computational complexity of each of the integration algorithms is not possible as it is a function of the particular image set. This depends on the number of images, m , the number of points in each image, N_k , the proportion of redundancy between images and the length of the boundary between overlapping images. A qualitative comparison of the worst-case computational complexity of each algorithm is given, for m images of N points, by approximating the cost of each stage of the integration process and deriving the overall order of complexity. To enable quantitative comparison of the computational complexity we consider a special case: the cost of integrating two images of N points with 50% overlap. Results of the complexity analysis are summarised in Table 1.

Nearest point search is common to all integration algorithms. Implemented as a brute force search the time complexity is $O(N)$, where N is the number of points. Uniform subdivision of the 3D space facilitates a local search for the nearest point, [4, 9]. Subdivision of the 3D space into voxels of approximately

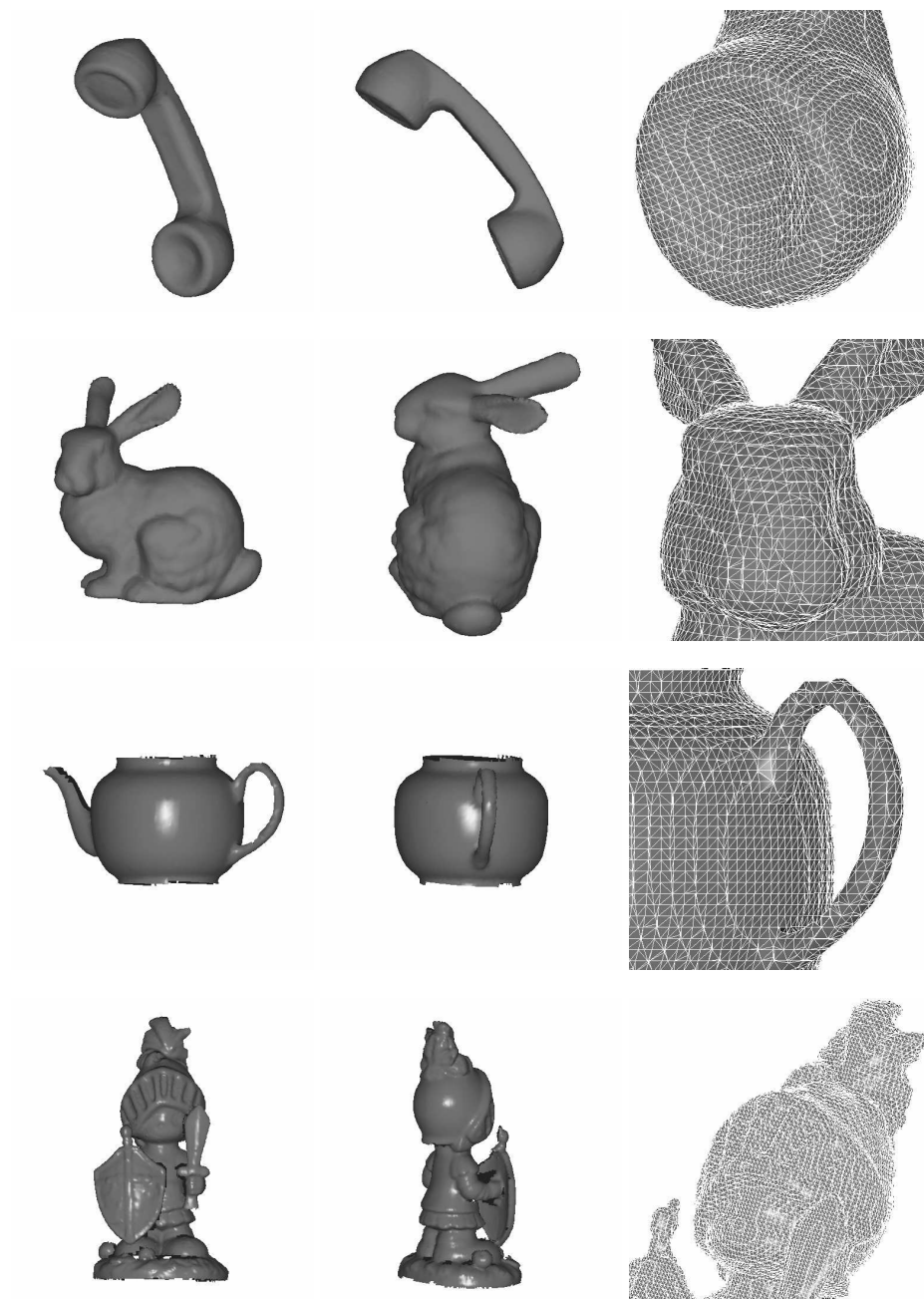


Fig. 2. Reconstructed 3D Models

Method	Overall Complexity m Images	Complexity of Integrating 2 images
I Implicit Surface Points+Normals	$O(m^2N)$	$6N$
II Implicit Surface Mesh	$O(m^2N)$	$12N$
III Canonic Views	$O(m^2N)$	$4N + 2\sqrt{N}$
IV Mesh Zippering	$O(m^2N)$	$N + \sqrt{N}$
V Mesh Growing	$O(mN \log N)$	$20N \log N (K = 10)$

Table 1. Integration Algorithm Complexity

the same size as the sampling resolution, $\Delta \mathbf{x}$, reduces the computational cost to approximately constant time. This can be efficiently encoded using a hash table representation requiring $O(N)$ space. This approach assumes that the individual range images are uniformly sampled and measurement error, $\varepsilon \ll \Delta \mathbf{x}$. Throughout this analysis it is assumed that nearest point search is performed in constant time. Method V requires the neighbourhood of K nearest points. A k -D tree provides a suitable data structure, [2, 4, 6]. Pre-computation of the kd-tree is $O(N \log N)$ complexity and space requirement is $O(N)$. The K nearest point search is then computed in $O(\log N)$ time.

Step discontinuity constrained 2D triangulation is common to methods (II—V). This operation consists of comparing the 3D position of each data point to its 8-neighbourhood and then thresholding to determine the connectivity. This requires on average 3 comparisons and Euclidean distance computations per data point. The step discontinuity constrained triangulation has computational complexity, $O(N)$.

Time complexity of the critical sub-stages for each integration algorithm are outlined below. This complexity analysis presents the algorithms in a common framework for qualitative comparison, [3]. It is assumed that the number of redundant points between two overlapping images of N points is $O(N)$ and that the resulting boundary length of non-redundant subsets is $O(\sqrt{N})$. Previously complexity analyses were presented for methods I [1, 4], III [8] and V [2].

Methods I and II: Implicit Surfaces

Computation of hash table spatial subdivision: $O(mN)$
Implicit surface function evaluation: $O(m)$
Number of marching cube implicit surface function evaluations: $O(mN)$
Overall complexity: $O(m^2N)$

Method III: Canonic views

Computation of Venn Diagram: $O(m^2N)$
Reparameterisation into canonic sub-views: $O(m^2N)$
Retriangulation of all redundant canonic subsets: $O(m^2N)$
Elimination of redundancy in canonic subsets: $O(m^2N)$
Retriangulation to build model: $O(m\sqrt{N})$
Overall Complexity: $O(m^2N)$

Method IV: Mesh Zippering

Redundancy test using nearest point: $O(m)$
 Elimination of redundant mesh elements: $O(m^2 N)$
 Clipping of mesh boundary: $O(m\sqrt{N})$
 Overall complexity: $O(m^2 N)$

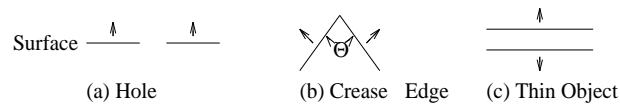
Method V: Mesh Growing

Computation of the k -D tree: $O(mN \log N)$
 Search for K-Nearest Neighbours: $O(K \log N) = O(\log N)$
 Surface retriangulation: $O(KmN \log N) = O(mN \log N)$
 Overall Complexity: $O(mN \log N)$

3.2 Geometric Limitations

This section identifies limitations inherent in each of the integration algorithms for reliable surface reconstruction. The comparison focuses on limitations on correct reconstruction of surface topology and geometry. Restrictions are identified by considering three cases: minimum hole size, maximum surface curvature and minimum surface separation (Figure 3). Results are summarised in Table 2. Table 2 also includes general characteristics which relate to the integration performance for a particular application. Computation type is specified as 2D or 3D according to the requirement for local or global projection to 2D sub-planes which imposes limitations on the local geometry. Static computation requires all data to be present prior to integration, conversely dynamic computation allows sequential addition of new data. Additionally some integration algorithms impose inherent limitations on the sampling process, restricting integration of range images at a single resolution. Further details are given in [3].

	2D/3D	Static/ Dynamic	Sampling	Minimum Feature Size	Minimum Crease Angle	Minimum Surface Separation
I	3D	static	uniform	$> 3n\Delta\mathbf{x}$	140°	$n\Delta\mathbf{x}$
II	3D	static	non-uniform	$n\Delta\mathbf{x}$	30°	ε_{max}
III	2D	semi-dynamic	non-uniform	$n\Delta\mathbf{x}$	30°	ε_{max}
IV	2D/3D	dynamic	uniform	$2n\Delta\mathbf{x}$	90°	$n\Delta\mathbf{x}$
V	3D	dynamic	non-uniform	$2n\Delta\mathbf{x}$	90°	ε_{max}

Table 2. Integration Algorithm Limitations for Reliable Reconstruction**Fig. 3.** Limitations on Surface Geometry

Minimum Feature Size: This section defines the minimum hole size that is guaranteed to be reconstructed. **Method I** does not explicitly represent mesh boundaries and therefore requires a threshold to be set for the maximum distance from the mesh, $\sqrt{3}n\Delta\mathbf{x}$. This results in invalid boundary extension of order $\sqrt{3}n\Delta\mathbf{x}$ which will fill any holes less than $2\sqrt{3}n\Delta\mathbf{x}$ in size. **Methods II-IV** use the local mesh connectivity for model reconstruction. The aim is to preserve holes that are identified in the step discontinuity constrained triangulation. The constrained triangulation imposes a lower limit on the surface feature size, $n\Delta\mathbf{x}$, for which the local surface connectivity will be reliably reconstructed. However, **method IV** may fail due to ambiguities between overlapping mesh elements that arise in the zippering algorithm. Holes less than twice the size of the individual mesh elements may redundantly overlap. This limits the minimum feature size for reliable reconstruction to $2n\Delta\mathbf{x}$. **Method V** does not use the local mesh continuity to constrain the topology of the triangulation. The maximum size of an element added to the triangulation is limited explicitly by a distance threshold. This threshold limits the minimum feature size for which the topology will be correctly reconstructed to $2n\Delta\mathbf{x}$.

Minimum Crease Angle: This section defines limitations on the reliability of the integration process for regions of high surface curvature and crease edges. This is quantified as the minimum crease angle, θ_{min} , between two intersecting planes across which the local topology is guaranteed to be correctly reconstructed. **Methods II-IV** may fail in regions of high curvature if the constrained triangulation does not correctly reconstruct the connectivity due to the increased distance between adjacent samples. The step discontinuity threshold, $t_d = n\Delta\mathbf{x}$, imposes a limit on the minimum crease angle, $\theta_{min} = 2\tan^{-1}(\Delta\mathbf{x}/n\Delta\mathbf{x})$. Typically $n \approx 4$ thus the sampling resolution imposes a minimum crease angle $\theta_{min} \approx 30^\circ$. **Methods IV and V** both require explicit thresholds to be set for the maximum difference in orientation between adjacent surface elements, $> 90^\circ$. This imposes a hard limit on the minimum crease angle that is reliably reconstructed, $\theta_{min} \approx 90^\circ$. **Method I** will fail if an incorrect tangent plane is used for the implicit surface distance function evaluation, $f(\mathbf{x})$. This results in spurious mesh artifacts in regions of high curvature. In practice this imposes a minimum crease angle for reliable reconstruction, $\theta_{min} \approx 140^\circ$.

Minimum Surface Separation: The third geometric structure that causes failure of the integration process occurs where different surface regions are in close proximity (i.e. thin object parts). This can be quantified as the minimum separation between different surface regions. Reconstruction will fail if the algorithm does not use the local connectivity and orientation information. **Methods II,III and V** use the local surface orientation to define overlapping regions. The lower limit for reliable reconstruction for these methods is therefore surface separated by the maximum measurement error ε_{max} . **Method I** does not use the local surface orientation and will therefore fail for surface regions separated by less than the sampling resolution, $n\Delta\mathbf{x}$. **Method IV** relies on a nearest point search to define the boundary intersection between meshes. This test may fail if meshes from different surfaces are closer than the sampling resolution, $n\Delta\mathbf{x}$.

4 Conclusion

A novel registered range image integration algorithm has been presented. This uses a mesh based implicit surface function to define the object surface as the zero-set of a field function defined at any point in 3D space. The aim of this algorithm is to estimate the underlying surface topological type from the measured data by reconstructing a triangulated model with the same topology. This is the first integration algorithm that operates entirely in 3D space. Reliable reconstruction is demonstrated for complex objects.

Performance characterization of existing integration algorithms is presented. Qualitative analysis of the computational complexity demonstrates that most algorithms are $O(m^2N)$ for integration of m images of N points. Quantitative analysis in specific cases indicates large difference in the constant time complexity associated with each method. Geometric limitations for reliable reconstruction are identified. This analysis demonstrates that reliable reconstruction is only guaranteed for the integration algorithms based on Canonic sub-views [8] and the mesh based implicit surface (this paper).

Future development of the integration algorithm presented in this paper should focus on dynamic data integration and computational efficiency. Further work on performance characterization is required to benchmark the relative computational cost and validate the geometric limitations identified.

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. A. Hilton. On reliable surface reconstruction from multiple range images. In *VSSP-TR-5-95*. <ftp://ftp.ee.surrey.ac.uk/pub/vision/papers/hilton-vssp-tr-5-95.ps.Z>, 1995.
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. M. Rioux. Laser range finder based on synchronized scanners. *Applied Optics*, 23(21):3837—3844, 1984.
6. 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.
7. M. Soucy and D. Laurendeau. Multi-resolution surface modelling from multiple range images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 348—353, 1992.
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.

This article was processed using the \LaTeX macro package with ECCV'96 style