Title:
Automatic Construction of Watertight Manifold Triangle Meshes from Scanned Point Clouds Using Matched Umbrella Facets

Authors:
Ji Ma, jima605@gmail.com, The University of British Columbia
Jack Szu-Shen Chen, jsschen38@gmail.com, The University of British Columbia
Hsi-Yung Feng, feng@mech.ubc.ca, The University of British Columbia
Lihui Wang, lihuiw@kth.se, KTH Royal Institute of Technology

Keywords:
Watertight Manifold Mesh, Point Cloud, Delaunay Triangles

DOI: 10.14733/cadconfP.2016.257-261

Introduction:

Due to the increasing applications of modern 3D scanning technologies, point clouds have emerged as an effective data source to generate the complete surface model of a scanned object. Converting a discrete point cloud data set into a triangle mesh surface representation, commonly known as mesh surface reconstruction, is one of the most widely employed approaches. One key requirement in the application field of computer-aided design and geometric modeling is for the constructed triangle mesh surface to be a watertight manifold surface with correct topology. However, due to the inevitable measurement noise in the scanned data as well as the surface quality of the original scanned object surface, it is still challenging to reconstruct a watertight manifold mesh surface that is topologically equivalent to the original object surface. An effective and reliable mesh surface reconstruction method is thus in high demand.

Existing surface reconstruction methods can essentially be classified into three groups: implicit surface, region growing, and Delaunay-based methods. Implicit surface methods [6],[8],[15] use the input point cloud to form a function that is formulated to be negative inside the modeled object surface and positive outside the modeled object surface. The desired object surface is extracted as the zero level set of the formulated function. In general, the implicit surface method can yield a watertight manifold mesh surface but the mesh only approximates the input point cloud without passing through the individual data points. For the region growing methods [9],[12],[13], the process begins with a seed triangle and then incrementally grows or expands from this seed triangle until the complete point cloud data set is covered. The outstanding issue with this group of methods remains the identification of an appropriate seed triangle. Region growing methods that grow the mesh by adding Delaunay triangles exclusively, have also been proposed [5],[10],[11]. For all the existing region growing methods, it is known that the reconstructed mesh much depends on the choice of the seed triangle and post-processing is often needed in order to attain a satisfactory watertight manifold triangle mesh. For the Delaunay-based methods [1],[2],[4],[7],[16], they aim to extract a subset of triangles from the complete set of Delaunay triangles when constructing the desired triangle mesh surface. The Delaunay-based methods are systematic and generally considered to be robust, but generating a watertight manifold mesh has consistently been a challenge. To address this issue, Amenta and her co-workers [2],[3] have introduced a Delaunay-based method with a theoretical guarantee for topological correctness of the generated mesh for points sampled from a smooth surface. The resulting triangle mesh is guaranteed to be topologically equivalent and geometrically close to the original smooth object surface if the particular sampling condition is met. The applicability of this condition to scanned point clouds with noise to construct a topologically correct mesh surface is, however, uncertain.

Proposed Method:

To address the outstanding issues of the existing methods, a new Delaunay-based method is presented in this paper, which improves on the preliminary Umbrella Facet Matching (UFM) method reported by the authors [14]. A flow chart of the overall algorithm is shown in Fig. 1. In essence, the UFM method seeks to iteratively generate, in parallel, a fully matched, local 2-dimensional manifold triangle mesh at each data point (resembling the shape of an open umbrella) from its Delaunay triangle set. An umbrella of manifold triangle facets is regarded as a fully matched umbrella when it fully overlaps with its neighboring umbrellas. Different from the method of Adamy et al. [1], once the algorithm converges and a fully matched umbrella at each data point is found, the generation of a watertight manifold triangle mesh is guaranteed without the need for additional mesh post-processing.

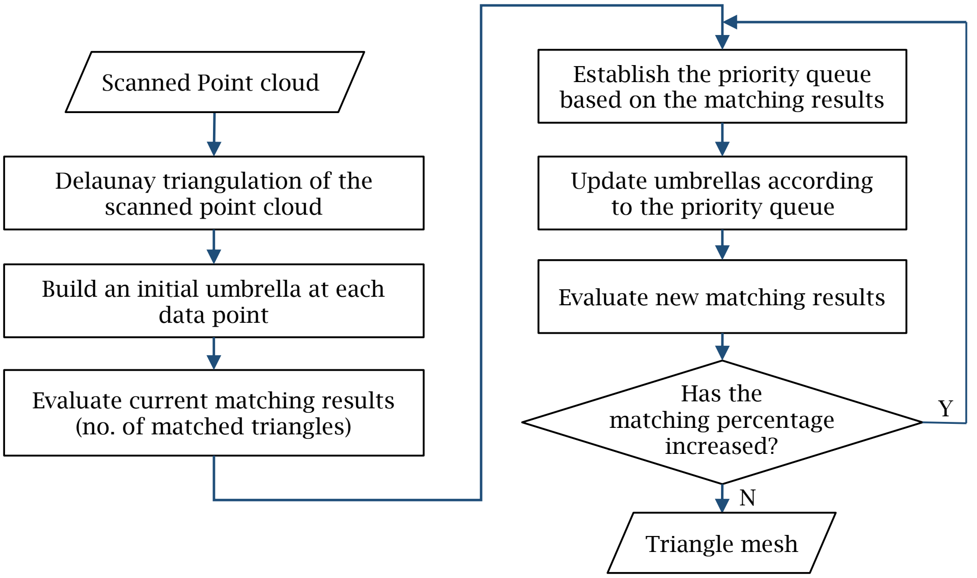
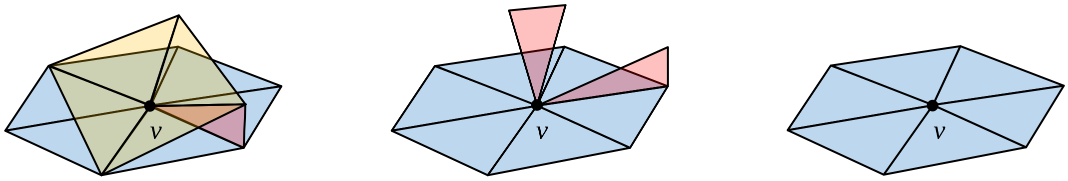


Fig. 1: A flowchart for the Umbrella Facet Matching algorithm.

Building an umbrella at each data point is basically a process that sequentially removes all redundant (non-manifold) triangle facets according to a priority queue. There are three fundamental topological types of Delaunay triangle clusters incident to a data point as shown in Fig. 2. A redundant triangle facet is either a non-manifold facet as part of a pocket of triangle facets, as highlighted in pink in Fig. 2(a), or a fin, as highlighted in pink in Fig. 2(b). The redundant triangle facet removal process starts with removing a non-manifold facet within a pocket, followed by a fin cleaning procedure. This facet removal process ends when there are no more non-manifold edges or vertices in the updated triangle facet cluster. The remaining triangle facets then correctly form an umbrella as depicted in Fig. 2(c).



(a) (b) (c)

Fig. 2: Three topological types of Delaunay triangle clusters: (a) umbrella with pockets, (b) umbrella with fines, and (c) manifold umbrella.

Proceedings of CAD'16, Vancouver, Canada, June 27-29, 2016, 257-261

It is evident that different priority queues for removing the triangle facets would lead to different umbrellas. This means that a specific priority queue at a data point has to be established in order to build the desired umbrella. Since the desired umbrellas are the fully matched umbrellas, the corresponding priority queue at each data point should be attainable by updating the priority queue according to the matching results of all the umbrella facets. For an existing umbrella, the matching results of its triangle facets are to be evaluated and then used to establish an updated priority queue via a priority queuing mechanism. The updated priority queue then leads to an updated umbrella. This process repeats until a fully matched umbrella is found. Hence, the priority queuing mechanism and how the umbrella facet matching results are evaluated and used to update the priority queue are clearly the core modules of the automatic mesh reconstruction algorithm in this work.

To construct the desired fully matched umbrella at every data point, a priority queuing mechanism with four-level inheritance is introduced, where a sub-level always inherits the queuing from a super-level. This means that the queuing rules should be prioritized and placed in an ordered sequence starting from the most superior level. For the priority queuing mechanism proposed in this work (Fig. 3), the queuing rule at the first (top) level is the absolute matching index M_f, representing the basic matching result; at the second level is the relative matching index $M_{f(v)}$, representing the refined matching result; at the third level is the neighboring matching index $M_{f(e)}$, representing the extended matching result; and at the fourth level is the size of the Delaunay triangle. In the initialization stage of building the initial/first umbrella, only the triangle size information is available (as no matching results exist yet). The diameter of the minimum circumsphere of the triangle is employed to quantify the triangle size. The initial priority queue is then established to remove redundant non-manifold triangles according to their sizes. If an umbrella cannot be constructed, the algorithm then resorts to the complete Delaunay triangle set to ensure that an initial umbrella is established at each data point.

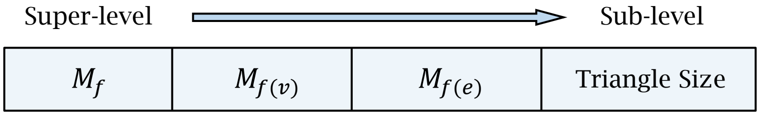


Fig. 3: Priority queuing mechanism with four-level inheritance.

After the initial umbrella at each point is established, three matching indices are evaluated to indicate the degree of overlap among the established umbrellas. The first two matching indices were introduced previously: basic and refined [14]. The basic matching result is quantified by the absolute matching index M_f and the refined matching result is quantified by the relative matching index $M_{f(v)}$. The absolute matching index M_f is devised to indicate the degree of matching for a facet f. The relative matching index $M_{f(v)}$ is devised to indicate the degree of matching for f relative to the vertex v. Let the three vertices of the facet f be v_1, v_2, and v_3. When M_f equals 3, this means that all of the three umbrellas respectively incident to v_1, v_2, and v_3 include the facet f. When M_f equals 2, only two of the three umbrellas respectively incident to v_1, v_2, and v_3 include the facet f. In this case, there exist two possible situations. Relative to the vertex v_1, one situation is that the umbrella at v_1 does not include the facet f and the other situation is that the umbrella at v_1 includes the facet f. Their relative matching indices are then respectively expressed as: $M_{f(v1)} = 0$ and $M_{f(v1)} = 1$. For the situation when all of the three umbrellas incident to v_1, v_2, and v_3 do not include the facet f, $M_f = 0$ and $M_{f(v1)} = 0$.

As stated previously, the presented UFM algorithm centers on the sequential removal of redundant triangle facets from the candidate triangle facet cluster. This is achieved via the priority queuing mechanism with multi-level inheritance according to the umbrella facet matching results. The priority queue is formed according to the evaluated values of the absolute matching index M_f followed by the relative matching index $M_{f(v)}$. More specifically, for all the triangle facets from the candidate facet cluster at v, the sequence is formed from $M_f = 0$ to $M_f = 3$ first and then from $M_{f(v)} = 0$ to $M_{f(v)} = 1$. As reported in the authors' preliminary work [14], there remain regularly many triangle facets incident to v which are characterized with the same $M_{f(v)}$. These facets were prioritized according to their minimum circumsphere radii in the same way as in the initialization stage. This sequencing much affects the convergence rate, if not the eventual convergence of the algorithm. To promote convergence of the proposed algorithm, an additional matching index has been introduced in order to enable triangle facets with the same $M_{f(v)}$ but larger circumsphere radii to be placed behind those with the smaller circumsphere radii in the priority queue of facet removal if so indicated by the current facet matching results. In other words, the removal priority of facets with the same $M_{f(v)}$ should not be simply based on their minimum circumsphere sizes. Current matched facets should carry decisive weights to promote neighboring facets that expedite the generation of fully matched umbrellas and the overall watertight manifold triangle mesh.

A neighboring matching index, $M_{f(e)}$, is employed as the third-level queuing measure in the priority queuing mechanism presented in this work (Fig. 3). As depicted in Fig. 4, there are three edges e_1, e_2 and e_3 in the triangular facet $f(v1, v2, v3)$ and each edge has some connected neighboring umbrella triangle facets. Each of these neighboring facets is characterized by an absolute matching index value from the current facet matching result. Let the maximum absolute matching index value among the neighboring facets of edge e_1 be denoted by M_{f1max} and its value can be 3, 2, 1 or 0. To evaluate the neighboring matching results of the facet $f(v1, v2, v3)$, the following index is devised:

$$M_{f(e)} = \sum_{i=1}^{3} M_{fimax} \tag{1}$$

where the value of $M_{f(e)}$ ranges from 0 to 9. For the triangle facets with the same $M_{f(v)}$ value, those with smaller $M_{f(e)}$ values (not much neighboring support) are to be placed in front of those with larger $M_{f(e)}$ values in the priority queue of facet removal. Those facets with the same $M_{f(e)}$ value are then ordered by their minimum circumsphere radii as the bottom-level rule in the priority queuing mechanism. Evidently, the priority queue at each data point will be continually updated until all the fully matched umbrellas are successfully found.

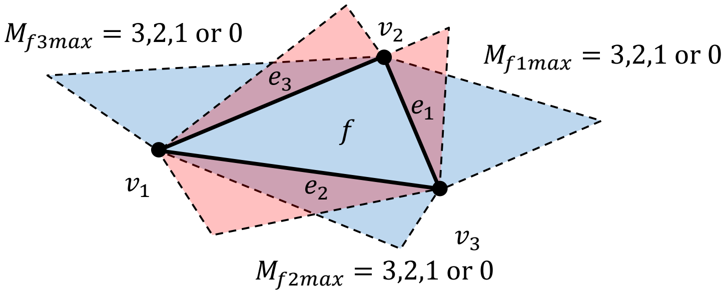


Fig. 4: Neighboring matching for facet $f(v1, v2, v3)$.

Typical Implementation Results:

By improving the priority queuing mechanism in the authors' preliminary work [14], an enhanced UFM algorithm to automatically generate watertight manifold triangle meshes from input scanned point cloud data sets has been developed and implemented. Typical generated meshes are shown in Fig. 5.

Fig. 5: Typical implementation results on scanned point cloud data sets.

These meshes are all topologically very close to the original scanned object surfaces with no influential or apparent mesh defects. The addition of the neighboring matching index and the resulting four-level inheritance priority queuing mechanism allow the enhanced UFM algorithm to better promote umbrella facet matching convergence, thereby improving the generated mesh quality and the mesh generation speed. From the series of case studies, it is found that the enhanced UFM algorithm in general outperforms the existing UFM algorithm in generating the water-tight manifold triangle meshes.

Acknowledgements:
This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References:
[1] Adamy, U.; Giesen, J.; John, M.: Surface reconstruction using umbrella filters, Computational Geometry, 21(1), 2002, 63-86. http://dx.doi.org/10.1016/S0925-7721(01)00040-2
[2] Amenta, N.; Bern, M.; Kamvysselis, M.: A new Voronoi-based surface reconstruction algorithm, Proceedings of SIGGRAPH '98, 1998, 415-421. http://dx.doi.org/10.1145/280814.280947
[3] Amenta, N.; Bern, M.: Surface reconstruction by Voronoi filtering, Discrete & Computational Geometry, 22(4), 1999, 481-504. http://dx.doi.org/10.1007/PL00009475
[4] Amenta, N.; Choi, S.; Kolluri, R. K.: The power crust, Proceedings of the 6th ACM Symposium on Solid Modeling and Applications, 2002, 249-266.
[5] Cohen-Steiner, D.; Da, F.: A greedy Delaunay-based surface reconstruction algorithm, The Visual Computer, 20(1), 2004, 4-16. http://dx.doi.org/10.1007/s00371-003-0217-z
[6] Curless, B.; Levoy, M.: A volumetric method for building complex models from range images, Proceedings of SIGGRAPH '96, 1996, 303-312. http://dx.doi.org/10.1145/237170.237269
[7] Dey, T. K.; Goswami, S.: Tight cocone: A water-tight surface reconstructor, Proceedings of the 8th ACM Symposium on Solid Modeling and Applications, 2003, 127-134. http://dx.doi.org/10.1145/781606.781627
[8] Dinh, H. Q.; Turk, G.; Slabaugh, G.; Reconstructing surfaces by volumetric regularization using radial basis functions, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(10), 2002, 1358-1371. http://dx.doi.org/10.1109/TPAMI.2002.1039207
[9] Huang, J.; Menq, C. H.: Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology, Computer-Aided Design, 34(2), 2002, 149-165. http://dx.doi.org/10.1016/S0010-4485(01)00079-3
[10] Kuo, C. C.; Yau, H. T.: A Delaunay-based region-growing approach to surface reconstruction from unorganized points, Computer-Aided Design, 37(8), 2005, 825-835. http://dx.doi.org/10.1016/j.cad.2004.09.011
[11] Kuo, C. C.; Yau, H. T.: A new combinatorial approach to surface reconstruction with sharp features, IEEE Transactions on Visualization and Computer Graphics, 12(1), 2006, 73-82. http://dx.doi.org/10.1109/TVCG.2006.2
[12] Li, X.; Han, C. Y.; Wee, W. G.: On surface reconstruction: A priority driven approach, Computer-Aided Design, 41(9), 2009, 626-640. http://dx.doi.org/10.1016/j.cad.2009.04.006
[13] Lin, H. W.; Tai, C. L.; Wang, G. J.: A mesh reconstruction algorithm driven by an intrinsic property of a point cloud, Computer-Aided Design, 36(1), 2004, 1-9. http://dx.doi.org/10.1016/S0010-4485(03)00064-2
[14] Ma, J.; Feng, H. Y.; Wang, L.: Delaunay-based triangular surface reconstruction from points via umbrella facet matching, Proceedings of the 6th IEEE Conference on Automation Science and Engineering, 2010, 580-585.
[15] Turk, G.; O'Brien, J. F.: Shape transformation using variational implicit functions, Proceedings of SIGGRAPH '99, 1999, 335-342. http://dx.doi.org/10.1145/311535.311580
[16] Xu, X.; Harada, K.: Automatic surface reconstruction with alpha-shape method, The Visual Computer, 19(7), 2003, 431-443.