Incremental Multiview Integration of Range Images

Angel Domingo Sappa †

LAAS-CNRS †
7, Avenue du Colonel Roche
31077 Toulouse, Cedex 4, France
asappa@laas.fr

Miguel Angel García ‡

Department of Computer Science and Mathematics ‡
Rovira i Virgili University
Ctra. Salou s/n. 43006 Tarragona, Spain
magarcia@etse.urv.es

Abstract

This paper presents a new method for the incremental integration of overlapped range images. It is assumed that frame transformations between all pairs of views can be reliably computed. This method proceeds by progressively merging each new sensed range image with the current reconstructed model. The proposed method consists of three stages. In the first stage the overlapped regions are identified. Then, from the overlapped regions, the points that define the integration boundaries are projected over a reference plane and are triangulated over that 2D space by means of a constrained Delaunay algorithm. Finally the obtained triangulation is backprojected to the 3D range image space. These new triangular meshes represent the sewing between the meshes to be integrated. In this way a new single triangular mesh which will be used to merge with futures range images is generated. Experimental results are presented.

1. Introduction

The automatic reconstruction of 3D objects (scenes in general) through range images is gaining popularity in computer vision and robotics owing to the variety of applications that can benefit from it, including world modeling, reverse engineering or object recognition.

This paper focuses on a part of the previous problem, which consists of the integration of range images obtained from different points of view. Those viewpoints may have been determined by applying a next-best-view algorithm, such as [1]. In particular, a new technique to incrementally integrate a set of range images into a non-redundant surface model is presented. In order to achieve this goal, each range image is described by a triangular mesh (e.g., [2]) and the result of the integration is also a triangular mesh which will be used to be integrated with subsequent range images.

Two different philosophies have been proposed in the literature to integrate multiple range images: *unstructured* and *structured* methods.

Unstructured integration methods are those which directly work with the original data points, generating a polygonal surface from that arbitrary set of unorganized points. The integration is performed, in this case, by collecting all the range points obtained from the different views. Following this unstructured approach, [3] presents an algorithm that integrates a set of 3D points that correspond to different views by applying a Delaunay triangulation algorithm. Following this line, [4] and [5] propose other techniques for multiview integration. The problem with these approaches is that all the range images to be integrated must be available as input from the beginning of the integration process.

On the other hand, *structured integration* methods work by using the original data arranged in some way (most of the structured methods assume that the different views to be integrated have been properly triangulated). Therefore, they can make use of topological and geometrical information associated with each point (e.g., point neighborhood, curvature, surface orientation). Turk et. al. [6] present the "zippering" technique to integrate range images represented by triangular meshes. This zippering technique, although works well for relatively smooth surfaces, has been shown to fail in regions of high curvature. Moreover, this technique has some problems when the triangles contained in the overlapped regions have the same coordinates. In this case, zippering can not find the intersection points between the edges of the different meshes. Soucy and Laurendeau [7] integrate a set of overlapped range images by decomposing them into subsets of *canonic views* (areas common to a unique subset of range images). This algorithm is extended in [8] allowing dynamic and sequential integration of new range images.

The proposed algorithm is enclosed within the structured integration methods and it is described in section 2. Section 3 gives experimental results by using synthetic range images. Finally, conclusions and further improvements are presented in section 4.

0-7695-0750-6/00 $10.00 © 2000 IEEE

2. Incremental multiview integration

Given a triangular mesh M_i , which approximates a certain range image, the objective consists of merging it with the current reconstructed model M, also represented by a triangular mesh, generating thus a non-redundant surface model that becomes the new current reconstructed model M.

The algorithm consist of three stages. The first stage identifies the overlapped regions between the two triangular meshes to be merged (the new range image and the current reconstructed model) and, by taking into account these overlapped regions, it defines the *integration boundaries*. Then, in the second stage, the integration boundaries are projected over an *integration plane*. The integration plane is a plane orthogonal to the *projection direction*, δ, computed by taking into account the orientation of the overlapped triangles. The points that constitute the projected integration boundaries are triangulated over that 2D space through a constrained Delaunay algorithm preserving in this way their connectivity. Finally the obtained triangulation is backprojected to the original 3D space. The final result is a non-redundant triangular mesh that will be integrated with further range images.

2.1. Overlap detection

Given two triangular meshes, M_i (new mesh) and M (current reconstructed model), referred to the same global coordinate frame, this first stage consist of the identification and removal of those triangles of M_i which overlap with some triangles of M, since they are considered to provide redundant information.

A triangle T_i of M_i is considered to be overlapped with the current model M if any of the vertices of that triangle are overlapped with any of their nearby triangles in M. A vertex of M_i will be considered overlapped with the triangle T if it belongs to either the upper or lower *overlap polytopes* of that triangle. The upper overlap polytope of a triangle is defined as the intersection between the positive half-space supported by the triangle's plane and the positive half spaces supported by three *upper overlap planes* associated with the edges of the triangle. Given an edge E of a triangle T, the upper overlap plane associated with E is the plane that contains E and is orthogonal to the plane of the triangle T' adjacent to T along E, if that neighbor exists, or orthogonal to the plane that contains T if T' does not exist or if triangles T' and T form a concave surface. Conversely, the lower overlap polytope is the symmetry of the upper overlap polytope with respect to the plane defined by triangle T.

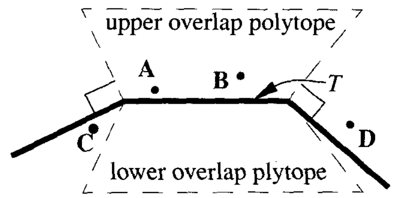

Figure 1. 2D illustration of the determination of overlap between a point of M_i and a triangle T of M. Overlap polytopes are shown in dark. Points **A** and **B** are overlapped while points **C** and **D** are not.

Fig. 1 illustrates the concept of overlap polytopes and overlap detection considering a 2D section of three triangles shown with a thickened polyline. In that example, vertices **A** and **B** are considered overlapped with the central triangle T while vertices **C** and **D** are not considered overlapped

Before eliminating the overlapped triangles of M_i , the *integration boundaries* of M are determined. The integration boundaries of M are defined by the edges that link the exterior vertices of M that are overlapped with triangles in M_i .

Next, those triangles of M_i with some of their vertices overlapped with M are considered overlapped and are eliminated. Thus, after removing all the overlapped triangles of M_i , some interior vertices of M_i are converted to exterior vertices. The edges which link these new sets of exterior vertices are considered as the *integration boundaries* of M_i.

Fig. 2 gives an example of two overlapped meshes and the results after eliminating the overlapped triangles. The integration boundaries are indicated with a thickened polyline.

2.2. Boundary projection and triangulation

At this stage, the integration boundaries found before are projected over a reference plane orthogonal to the projection direction δ. Vector δ is computed as the average direction of the normal vectors associated with the triangles of M which have some of their edges belonging to the integration boundaries.

The set of points that defines the projected integration boundaries are triangulated by means of a constrained 2D Delaunay algorithm. The constraints of the triangulations are the edges corresponding to the projected integration boundaries. The result of this triangulation is a 2D triangular mesh whose convex hull encloses those projected boundary (Fig. 3 *top*).

Finally, the boundary edges (exterior edges) of the 2D triangulation that do not correspond to integration boundaries are removed, except for those cases in which an

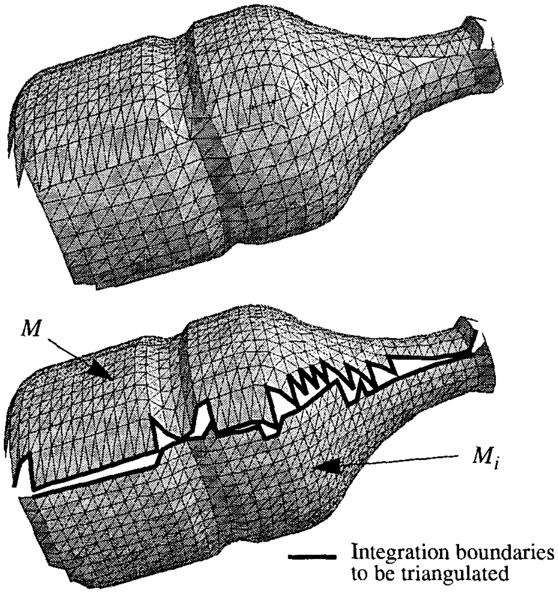

Figure 2. (*top*) Overlapped triangular meshes, representing the range images to be integrated. (*bottom*) Obtained representation after eliminating the overlapped triangles of M_i.

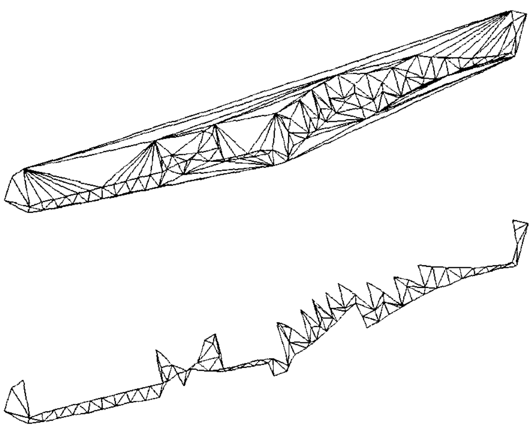

Figure 3. (*top*) 2D triangular mesh generated with the projected integration boundaries of M and M_i. (*bottom*) Triangular mesh obtained after removing all the boundary edges that are not integration boundaries.

edge joins a vertex of the integration boundary of M with a vertex of the integration boundary of M_i. Fig. 3 illustrates the previous triangulation and removal stages.

After obtaining a triangular mesh whose boundary edges are integration boundaries, a backprojection stage is responsible for mapping each 2D triangle to the original 3D space.

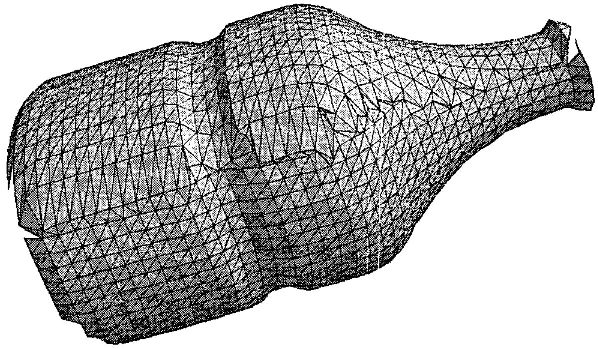

Figure 4. Final result obtained after integrating M and M_i. The resulting triangular mesh will be used to be integrated with further new range images.

2.3. Triangle backprojection

In this final stage, the obtained triangular mesh is represented in the original 3D space, merging in this way the triangular meshes to be integrated. This stage simply maintains the topology generated in the 2D projection. The triangles are backprojected to the 3D space and the integration process is ended. Fig. 4 shows the results of integrating the two triangular meshes used as an example throughout this paper.

3. Experimental results

The proposed technique has been tested with different range images obtained with a 3D graphical simulator. This tool allows the set-up of 3D scenes with arbitrary objects imported from CAD and from real range images. The system allows the definition of camera positions and the computation of range images with different resolutions. The simulator also provides the necessary transformation matrices between the local coordinate frames attached to the camera and the global observation frame. Each range image correspond to the digitalization of the object viewed form that sensor position.

Fig. 5 shows an incremental integration of four range images, each one is represented by its corresponding triangular mesh. Fig. 5 (*top*) shows the resulting triangular mesh obtained after the integration of the first two range images, with 743 and 629 vertices respectively. The CPU time to compute the three stages of the whole integration algorithm was 0.627sec. The resulting triangular mesh contains 1,218 vertices; this triangular mesh was integrated with a third range image. The latter was approximated with a mesh containing 470 vertices. The result of the integration is shown in Fig. 5 (*middle*). The CPU time to compute the integration process was 0.538 sec, and the obtained triangular mesh contains 1,505 ver-

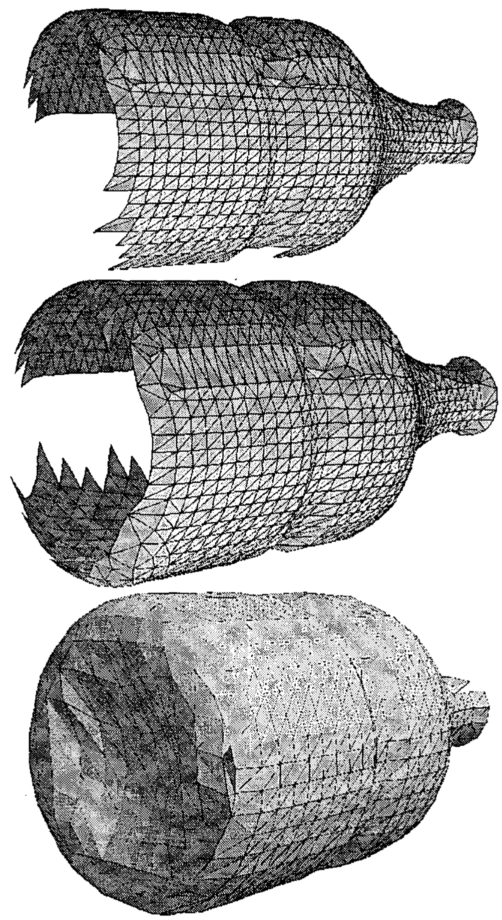

Figure 5. (*top*) Triangular mesh obtained after integrating two views. (*middle*) Addition of a new view (a new M_i mesh) to the previous result. (*bottom*) A fourth view is integrated with the previous mesh, which represents the integration of the three precedent meshes.

tices. Finally, Fig. 5 (*bottom*) shows the result when a fourth range image, of 521 vertices, is added. This last integration took 0.693 sec. The resulting triangular meshes are shown from different points of view. CPU times have been measured on a SGI Indigo II with a 175 MHz R10000 processor.

4. Conclusions and further improvements

A new incremental approach for the integration of overlapped range images approximated by triangular meshes has been presented. First the overlapping regions are detected and the integration boundaries are defined. Then, a projection direction is computed as the average of the normal vectors associated with the overlapped boundary triangles of the current model. By using this vector direction, the integration boundaries are projected over a reference plane. The vertices that define these boundaries are triangulated over that 2D space by means of a constrained 2D Delaunay algorithm, which considers the edges of these boundaries as constraints. Finally, the obtained triangulation is backprojected to the original 3D space, integrating in this way the two triangular meshes. The result is also a triangular mesh which can be used for further integrations.

The proposed technique is advantageous with respect to previous proposals due to the fact that it allows a continuous addition of information (range images) to the model which is being reconstructed. Moreover, this proposal is more efficient than previous ones as it is not necessary to compute 3D edge intersections or plane intersections. Finally, the original topology is not modified.

We are currently studying the application of some geometric techniques, such as NURBS [9], and convex interpolation functions [10], in order to take advantage of the redundant information present in the overlapped region. The aim of the application of these techniques is the reduction of the uncertainty introduced by the sensors over those overlapped regions.

5. References

[1] M. A. García, S. Velázquez and A. Sappa, A two-stage algorithm for planning the next view from range images, *British Machine Vision Conf.*, Southampton, UK, pp. 720-729, September 1998.

[2] M. A. García, A. Sappa and L. Basañez, Efficient approximation of range images through data dependent adaptive triangulations, *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 628-633, June 1997.

[3] J. Boissonnat, Geometric structures for three-dimensional shape representation, *ACM Trans. on Graphics*, vol. 3, No. 4, pp. 266-286, October 1984.

[4] H. Edelsbrunner and E. Mücke, Three-dimensional alpha shapes, *ACM Trans. on Graphics*, vol. 13, No. 1, pp. 43-72, 1994.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, Surface reconstruction from unorganized points, *Proc. Computer Graphics*, pp. 71-78, July 1992.

[6] G. Turk and M. Levoy, Zippered polygon meshes from range images. *SIGGRAPH'94*, 311-318.

[7] M. Soucy and D. Laurendeau, Multi-resolution surface modelling from multiple range images, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 348-353, 1992.

[8] M. Soucy and D. Laurendeau, A dynamic integration algorithm to model surfaces from multiple range views, *Machine Vision and Application*, vol. 8, pp. 53-62, 1995.

[9] G. Farin, *Curves and surfaces for Computer Aided Geometric Design*, academic Press, 1993.

[10] M. A. García, Reconstruction of visual surfaces from sparse data using parametric triangular approximants, *IEEE Int. Conf. on Image Processing*, Austin, USA, 750-754, 1994.