SURFACE SEGMENTATION USING A MODIFIED BALL-PIVOTING ALGORITHM

Andrés Restrepo Specht   *Michel Devy*

LAAS-CNRS
7, avenue du Colonel Roche
31077 Toulouse, France
E-mail: {arestrep, michel}@laas.fr

ABSTRACT

Ball Pivoting Algorithm (BPA) was originally made for computing a triangular mesh from a regular distributed cloud of points after a registration process. In this article this method is presented and adapted for the mesh creation of a non-regular distributed surface of points and the simultaneous surface segmentation. Results are obtained from 3D data acquired by several sensors: laser range finder, stereo or profilometer.

1. INTRODUCTION.

In the last years, a lot of work has been made on 3D objects or environment modelization. The applications that need these type of models are numerous, in addition after the developments in virtual reality (remote controlled processes, video games, virtual visit of touristic sites or museums, building or town modelization). Our interest is mostly centered in robotics.

In previous articles, we have presented some works dealing with the 3D modelization tasks, registration between 3D images [1, 2] and incremental construction of triangular meshes [3]. In this article, we suppose that the problem of registration is solved and we are interested in the mesh construction or the set of surface primitives in a 3D cloud of points. Therefore, we propose to adapt the algorithm given by Bernardini, Mittleman, Rushmeier, Silva and Taubin, the *Ball Pivoting Algorithm* (BPA) [4]. The original method deals with the construction of triangular meshes. Our BPA adaptation intends to make the algorithm more robust to registration errors and measurement noise and achieves the surface segmentation at the same time as the mesh is created, taking advantage of the "region growing" behavior of the algorithm.

This article is composed of 4 sections: the section 2 presents a short state of the art in the mesh construction methods. The sections 3 and 4 are devoted to the description of the method and the result presentation on different types of 3D images. At the end, in section 5 we will present the conclusions and evoke our actual works.

2. STATE OF THE ART.

The technological advances in 3D data acquisition systems have motivated the conception of different methods for object reconstruction. These methods are adapted to the application properties (type of objects, resolution, 3D model precision) and to the acquisition methods. An object or an environment can be represented in many forms, but triangular meshes were imposed thanks to their generality. The incremental construction techniques for triangular meshes, based on 3D points, can be classified in two big categories:

Iterative subdivision-based methods: they use typically the Delaunay triangulation, which gives a theoretical guarantee for the final mesh quality. The disadvantages are the amount of memory needed and the computational time and also the numerical instability.

Region growing-based methods: the principal advantage is the robustness to acquisition errors, but on the other side these methods need a seed triangle to start with the surface creation process.

More explicitly, Hoppe and DeRose [5] have proposed a subdividing method, considering the distance function between each point and a tangent surface, while Bernardini and Bajaj [6] use the distance function in the alpha form of the cloud of points. Attali [7] make the surface reconstruction using a subset of the Delaunay diagram. Boissonnat and Cazals [8] applied a hybrid technique, with points and surface normals, calculating the surface as a set of zeros of a function, which gives a natural neighborhood relation between points. In [3], we deal with the incremental mesh construction using *Split* techniques; one of the mayor difficulties consists in the detection of common parts in the actual mesh, created with the preceding images, and the local mesh constructed with the actual image, taking into account the possible occultations.

The following works made similar choices in the method presented in this article. Fisher, Fitzgibbon and Eggert [9] perform a point cloud tessellation according to the Hoppe and DeRose method, then execute a classification of the local geometric form of the object to be modelized, in order to adapt the model construction by region growing. Bernardini, Mittleman, Rushmeier, Silva and Taubin [4] developed the BPA triangulation method; it's an iterative model based on alpha forms, allowing to merge in a mesh previously registered data. This is the method that we have adopted, since it can handle a great number 3D points with little calculation cost. We have introduced two modifications: 1) Treat the 3D images incrementally, therefore merge each image after being acquired and registered with the mesh in construction. 2) Simultaneous segmentation of the model in a set of surface primitives (here planes).

3. BPA ALGORITHM DESCRIPTION.

The BPA process starts with a *seed* triangle, whose three points belong to a sphere with radius ρ chosen by the user; see figure 1. This ball pivots around a triangle edge, always maintaining contact with the two vertices of the pivot edge, until it finds another measurement point, used to form a new triangle (figure 2), or if no

0-7803-8554-3/04/$20.00 ©2004 IEEE.

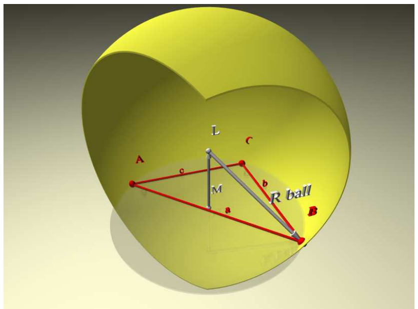

Fig. 1. The pivot ball with radius ρ and center L, lay on a triangle ABC. M is the center of the circumscribed circle to ABC.

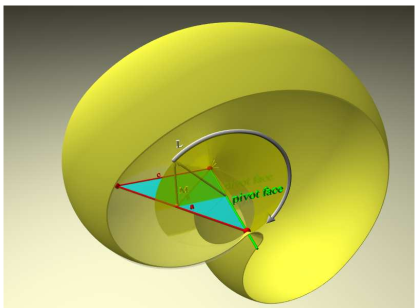

Fig. 2. The pivot ball around the BC axis, which is the designated pivot edge.

point is found, the edge will be marked as a frontier edge of the surface. The process will be repeated with an edge of the new triangle or on a free edge of another already created one. If all edges have been analyzed, a new seed triangle must be selected between the not used measurement points and the process can start again. The process stops when all edges have been checked and all points have been analyzed.

In our BPA version, the original method [4] has been modified to segment simultaneously the surfaces and to be more robust for variable resolution images. Some term definitions to be used are:

seed triangle: it is the triangle from which the surface mesh is generated.

pivot triangle: is the triangle used by the BPA to look for new triangles. The first pivot triangle is the seed triangle; in general, the created triangle of the previous iteration is the actual pivot triangle.

pivot edge: is the edge of the pivot triangle, around which the ball is turning to find new triangles.

frontier edge: it's an edge without any adjacent triangle, so it's on the frontier of the surface.

3.1. Initialization: seed triangle selection.

A crucial step in the process is to find a good seed triangle to start the method, once the pivot ball radius is given. The selection of the radius must be adapted to the point cloud resolution to be meshed, on the other side this radius will set the quality of the built model. If the radius is too small, the ball can't include the seed triangle, or it can't find any measure points pivoting around the available edges, which will turn to be frontier edges of the surface. The model will not be continuous and will have many holes. On the other side, if it's too large, the ball will find too much points; but only one can be chosen to construct the next triangle, while the rest will be marked as invalid. As the chosen point must be in contact with the pivot ball, the distances between the selected points will be about the ball radius, therefore the final model will be continuous but the form will not be as faithful as the original cloud of points.

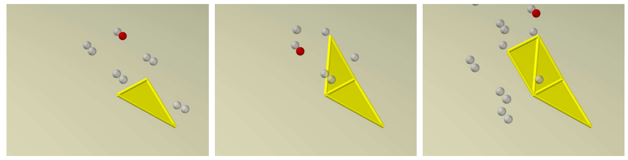

Fig. 3. The triangle construction step by step: (left) seed triangle and candidate points found by the ball rotation around the pivot edge; the dark point is the selected point. (center) second triangle, (right) third triangle.

3.2. The triangulation process.

The triangulation process for a surface starts with the seed triangle and continues until there are no more integrable points on this surface. If free points remain, a new seed triangle will be selected and a second surface will be created, until the end of all free points. The figure 3 illustrates the creation of the three first triangles of a surface. The steps are explained below.

3.2.1. Candidate point exploration.

Starting with a pivot edge, a base task consists in finding the candidate points for the vertex of the next triangle in the cloud of points. The point scan routine takes advantage of a *voxel map*, with a resolution proportional to ρ, to optimize the CPU time; only the voxels around the pivot triangle are analyzed.

The point selection criteria are applied in the following classification steps. At first, all the points that are inside the pivot ball trajectory around the pivot edge are selected. Afterwards, for every candidate point that would form a triangle with the two points of the pivot edge, the normal, area and angles are calculated. The mean normal of the surface is used to found the candidates on the surface. At last, for the candidate points that made it through this step, the formed candidate triangle is checked for intersection with the existing triangles of the voxel neighborhood.

At the end, we have a set of free and frontier candidate points from which the best two ones have to be chosen. This best candidate selection is based on a triangle quality test always trying to achieve equilateral triangle proportions.

3.2.2. Selection of the pivot edge.

In the most cases, the pivot triangle has two frontier and one occupied edges, being the last one the pivot edge of the preceding iteration, which is in contact with the surface in construction. These two edges are checked out at the same time to choose the best to be the next pivot edge.

It starts to look on the edge that is oriented in the same manner as the edge of the precedent iteration. After examination of the candidate points of the edge 1, if there is another edge available, it will be scanned only if no points were found on edge 1 or if there are no frontier points between the candidates of edge 1. Once the second edge has been analyzed, the selection between the two edges depends from the frontier points found on each edge.

In the worst case, for each edge, the candidate points selection will give two candidate points: one free point and one frontier point. Therefore, for a "normal" pivot triangle with two frontier edges , we will have a pair of best candidates, 4 points in total: two free and two frontier points, which now will be compared to choose the actual pivot edge.

This treatment seems complex, but it creates a mesh with a spiral-growing tendency around the seed triangle, which lets reduce the hole creation. The frontier point is generally preferred in the selection, even if the resulting triangle has lower quality.

3.2.3. New triangle creation.

The selected point forms with the pivot edge vertices a new triangle, which will become the next pivot triangle. The *new pivot edge* has by default the same reference as the previous one. If the edge is marked as not-free, it must be changed. If the new triangle has no frontier edges, no further growing is possible, then a new pivot triangle on the same surface has to be chosen from the list of created triangles, looking for free edges. If there are no more free edges, the process stops for this surface. Another seed triangle must be chosen from the free points. The process ends if there are no more free points available.

3.3. Pseudo-code of our BPA method.

```
TS=Seed_Triangle_Selection();
pivot_edge=search_edge(TS);
create_surface();
while(TS != 0) {
  No_candidates = Search_points(pivot_edge);
  if(No_candidates != 0)   { Select_candidat();
                             Add_triangle(TP);
                             update_structure();
                             pivot_edge=search_edge(TP); }
  if( (pivot_edge == 0) || (No_candidates == 0) ) {
      TP=Select_pivot_triangle();
      if(TP == 0) { close_surface();
                    TS=Seed_Triangle_Selection();
                    pivot_edge=search_edge(TS);
                    create_surface(); } } }
```

3.4. Incremental mesh construction or improvement.

The algorithm has been modified so that it is possible for the user to start the process with an existing mesh. This aspect is also useful for the creation of a variable resolution mesh, a mesh made with patches created by different radius pivot balls. It helps to solve some intrinsic problems of the algorithm philosophy, more precisely the tendency to create holes. The BPA algorithm is applied iteratively starting from an existing mesh, selecting a frontier edge as the pivot axe and using a greater ball radius in the hope to "catch" an already created triangle vertex or a free point, in order to close a hole.

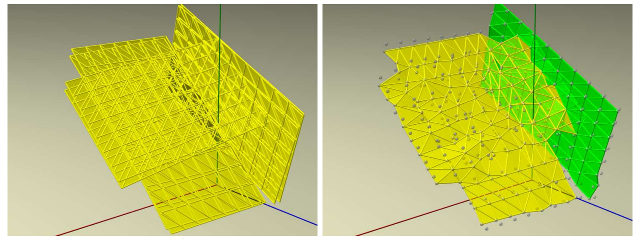

Fig. 4. Result of BPA with segmentation (right) on a synthetic cloud composed of six views of two orthogonal planes (left)

4. RESULTS AND DISCUSSION.

The BPA has been tested on different types of data, with or without polygonal meshes, only points and also on a set of post-ICP surfaces:

Synthetic Images: we have generated synthetic polygonal surfaces with uniform distributed points, in order to evaluate the basic behavior of the BPA algorithm. The measure and registration errors has been simulated mounting multiple views together. Every time in these examples, the BPA use the local mesh of the different views to look for the seed triangle. The figure 4 present six views of the two orthogonal planes. They are also lightly mounted one over the other, again, in order to evaluate the segmentation properties of BPA. The two expected surfaces has been detected.

Mesh after an ICP registration: The results of a registration process with ICP have been used to evaluate the BPA algorithm behavior producing a new mesh after an ICP. We show here a synthetic and a real example. In the figure 5, 9 synthetic meshed laser sensor views of a geometric object has been used. All 9 views have been previously decimated to reduce the number points. Afterwards the images has been registered with ICP in pairs (introducing even more errors into the registering process, being an incremental registration preferable) forming the complete cloud of points. The final ICP result gives a non uniform distributed cloud of points with overlapping meshes. The BPA detected successfully every object surface and took the needed points for the new mesh. The figure 6 shows a cloud composed of two 3D images, which are dense stereo reconstructions. In the same manner, these images have been decimated before the registration with ICP. The BPA has tried to produce a new mesh and simultaneously a segmentation in surface primitives. The point distribution of this type of image is locally dense, but not uniform, in particular on the soil. The best strategy is to execute BPA iteratively starting with a little ball radius and increasing it at every iteration.

Profilometry images: Images from a profilometry for human body modelization have been used: The profilometry acquisition method gives as output sequences of circular horizontal point scanlines, where the horizontal distances between scan points inside a scanline and the vertical distances to the other scan-lines are different. There is no local mesh available, therefore the seed triangle for BPA is extracted from the points of the cloud.

4.1. Performance.

The CPU times for the the chosen examples are given for a Pentium III 933MHz; the results are presented in the table 1.

Example	Points	Initial Polygons	Ball radius	Final Polygons	Segments	CPU time [s]
Stereo	8561	15998	0.31	5250	6	138.00
geometric object	6362	10838	0.3	1919	9	86.00
profilometry 1	9356	0	7	4247	24	76.00
profilometry 2	5952	0	7	3428	16	64.00
synth-6planes	480	756	0.47	348	2	2.00
synth-2planes	320	504	0.47	135	1	0.92

Table 1. Results of the BPA process on different types of data

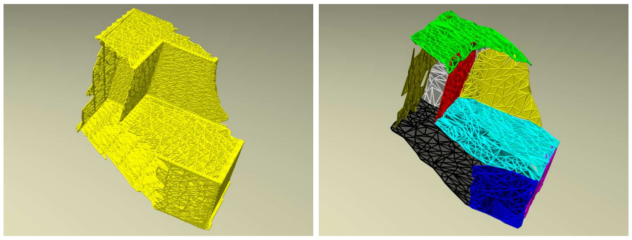

Fig. 5. BPA results (right), with construction of a mesh and segmentation, on 9 registered views acquired from a polyhedric object (left). The faces given by BPA are illustrated in false color.

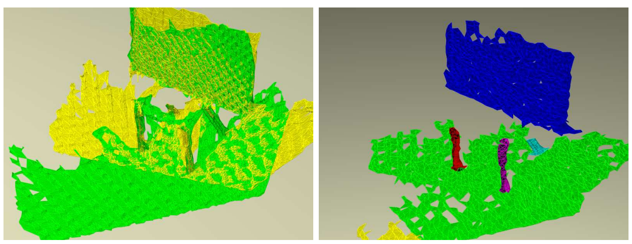

Fig. 6. BPA results (right), with construction of a mesh and segmentation, on 2 registered views acquired on a scene with dense stereo (left).

5. CONCLUSIONS.

The triangulation algorithm behaved as expected in [4], for the regular distributed cloud of points, found typically in laser sensor and profilometry images. For variable distributed clouds of clouds, our proposed modifications on the method gave good results. The mesh recycle feature of BPA proves to be very useful in these cases.

The segmentation in surface primitives, in planar and regular curved surfaces, gave acceptable results for structured scenes.

As observed in the tests on the images, we can conclude that our adapted version of the BPA algorithm is robust.

At last, as we have done in our precedent works [3], the algorithm can be optimized using information given by the 3D edge extraction, so as to impose the vertex presence over the discontinuities of the scene to be modelized. The edges can be extracted very efficiently either analyzing line by line and column by column each acquired 3D image on the scene [10, 11] , or exploiting a stereovision image by segment pair matching in order to extract the 3D segments.

6. REFERENCES

[1] A. Sappa, A. Restrepo Specht, and M. Devy, "Range image registration by using an edge-based representation," in *Proc. 9th International Symposium on Intelligent Robotic Systems (SIRS'2001), Toulouse (France)*, July 2001.

[2] M. Devy, A. Restrepo Specht, and A. Sappa, "Comparaison de méthodes de recalage de données 3d par construction d'un maillage triangulaire ou extraction des contours," in *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2002), Angers (France)*, Jan. 2002.

[3] J. Bozier, M. Devy, and A. Sappa, "A geometrical approach for the incremental modelling of free form surface by triangular meshes," in *8th Int. Symp. on Intelligent Robotic Systems (SIRS'2000), Reading (UK) , pp.13-21, Rapport LAAS N.00332*, July 2000.

[4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.

[6] C. L. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3D scans," *Computer Graphics*, vol. 29, pp. 109–118, 1995.

[7] D. Attali, "r-regular shape reconstruction from unorganized points," in *Symposium on Computational Geometry*, 1997, pp. 248–253.

[8] J.D. Boissonnat and F. Cazals, "Smooth surface reconstruction via natural neighbour interpolation of distance functions," in *Symposium on Computational Geometry*, 2000, pp. 223–232.

[9] R. Fisher, A. Fitzgibbon, and D. Eggert, "Extracting surface patches from complete range descriptions," *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa*, 1997.

[10] A. Sappa and M. Devy, "Fast range image segmentation by an edge detection strategy," in *Proc. Third International Conference on 3D Digital Imaging and Modeling, Quebec (Canada)*, May 2001.

[11] X. Jiang and H. Bunke, "Edge detection in range images based on scan line approximation," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 183–199, Feb. 1999.