The Ball-Pivoting Algorithm for Surface Reconstruction

1. Briefly summarize the paper's contributions. Does it address a new problem? Does it present a new approach? Does it show new types of results?

- **[AS]**
  This paper presents a new approach to the existing problem of data integration, or generating a surface representation from a given set of sample points. The approach presented in this paper is called the ball-pivot algorithm (BPA), which is a region growing method, resulting in a triangle mesh. This mesh is created roughly, ignoring details, by starting with a seed triangle, pivoting a ball around each edge on the existing mesh boundary, adding new triangles defined by this edge and a point hit by the ball, and repeating with new edge for pivoting. It is simple, efficient in terms of both execution time and storage, and robust to noise in real 3D scanned data.
- **[DS]**
  Presents a conceptually simple algorithm that is efficient in terms of time (linear) and storage.
  It presents a new approach to the problem of interpolating a mesh from a point cloud (especially obtained from scanning data).
  The algorithm is also robust to 3D scanning noise, so it can be used for practical applications.
- **[FP]**
  It proposes an algorithm for efficiently computing a triangular mesh formed by alpha-exposed triangles of an alpha-shape. Since it acts locally, it can be implemented by partitioning the input in slices (reducing memory requirements) and it is able to generate meshes from large data sets (e.g., *Pieta*).
- **[JD]**
- **[LF]**
  The paper presents BPA which is a new approach for finding the triangle mesh of an unorganized set of points that actually lends itself to brief explanation. Beginning with any three points, i.e. a seed triangle, a "ball" of a user specified radius pivots around each pair of points until it hits another sample point and that sample point is added to the mesh. That process is repeated on a new edge, formed between one of the pivot points and the new discovery. And so on, until we have our bounded mesh.
- **[MK]**
  The paper proposes an approach for computing an alpha shape in linear time. Because the value of alpha is fixed, it supports a streaming implementation.

2. What is the key insight of the paper? (in 1-2 sentences)

- **[AS]**
  The key insight of this paper is that a simple method such as BPA can quickly and robustly produce meshes that are subsets of alpha shapes of points sets.
- **[DS]**
  By pivoting a ball of a fixed radius around different edges in a point cloud, a mesh can be efficiently reconstructed. The algorithm adds a triangle when the ball encounters a new vertex as it pivots.
- **[FP]**
  A region growing algorithm to compute a mesh from alpha-exposed triangles can be efficiently implemented by pivoting a fixed radius ball over triangle edges.
- **[JD]**

- **[LF]** I probably do not know enough about surface reconstruction yet to discern what is truly insightful, but it seemed interesting that the method would sort of simultaneously correct sampling errors while aligning the meshes.
- **[MK]**
  The key insight of this paper is that if we already now a triangle on the alpha hull, we can compute the edge adjacent triangle by "pivoting" around the edge. Furthermore, the new vertex has to be within a distance of 2*alpha of a point on the edge, hence it suffices to create a regular voxel grid with bins the size of 2*alpha and just search for nearest neighbors.

3. What are the limitations of the method? Assumptions on the input? Lack of robustness? Demonstrated on practical data?

- **[AS]**
  One of the drawbacks of this method is that it assumes that the samples are distributed over the surface of the scanned object with a spatial frequency greater than or equal to some specified minimum value, and that an estimate of the surface normal at every point is available. However, these assumptions are valid for many acquisition techniques.
- **[DS]**
  Assumptions:
  -- Samples are distributed over the entire surface with frequency greater than or equal to an application-specified value.
  -- An estimate of the surface normal is available at each sample.
  The result might depend on the order of the choice of seed triangle and order of computation.
  This approach has been shown to run in linear time for datasets with millions of input samples.
- **[FP]**
  It requires uniform sampling in order to generate the mesh from a fixed radius ball. The algorithm also requires having an estimate of the normal at every point. Since the algorithm is interpolative and acts locally it seems very susceptible to produce undesired reconstructions in moderate noise regions (they alleviate this by preprocessing the data). Finally, the algorithm seems to be order dependent, this means, different outputs may be obtained for different orders of processing edges or picking seed triangles.
- **[JD]**
- **[LF]**
  The method makes two assumptions about samples: (1) samples are distributed over the entire surface with some frequency greater than a specified minimum and (2) a surface normal is available for each sample.
  The method appears to be robust and has been demonstrated on large datasets, particularly scans of the Michelangelo's Florentine Pieta and some examples from the Stanford 3D model set.
- **[MK]**
  For the method to work, as with alpha shapes in general, we require that the sampling density be at least alpha. Additionally, for the method to achieve its linear time (sub-linear space) complexity bounds, we also require that point sampling not be too dense. The method also assumes that normals are given.

4. Are there any guarantees on the output? (Is it manifold? does it have boundaries?)

- **[AS]**
  The method guarantees not only that the output is a manifold, but also that it is a subset

of an alpha-shape of the point set. Hence, it exhibits some nice properties of alpha-shapes, such as provable reconstruction guarantees under certain sampling conditions.

- **[DS]**
  The output is an interpolating triangle mesh.
  Reconstructs a surface homeomorphic to and within a bounded distance from the original manifold.
- **[FP]**
  The output is always a manifold (i.e., locally a disc in R^2). They also present two conditions on sampling and the radius of the pivoting ball (rho) that guarantee homeomorphism between the subjacent surface and the triangular mesh:
  a) The surface must be densely sampled: at least a sample in any rho-ball centered at the surface center.
  b) The surface should not curve in such a way those far regions in surface distanceget close in space distance: any rho-ball intersects the surface in at most a disk.
  The ball pivoting algorithm also produces boundaries.
- **[JD]**
- **[LF]**
  The method guarantees that the output will have a continuous, reversible mapping to and be within a bounded distance from the actual manifold.
- **[MK]**
  The output is guaranteed to be a non self-interesecting, possibly with boundaries, with manifold edges. It was unclear if the vertices were also guaranteed to be non-manifold.

5. What is the space/time complexity of the approach?

- **[AS]**
  The method described exhibits both efficiency in execution time and space requirements. The algorithm runs in linear time, linear in the number of sample points. It also uses linear storage.
- **[DS]**
  Runs in linear time.
  The resulting mesh is saved to external storage, so no additional storage is required.
  Out-of-core extension discussed to further reduce space requirements.
- **[FP]**
  It is linear time and linear storage under the assumption of bounded data density. In such case, each edge is visited once and ball pivoting evaluates a finite number of point candidates. On finding a seed triangle it evaluates each vertex at most once, using finite number of pairs to complete the triangle vertices, and doing the ball–interior test with a finite number of neighbors.
- **[JD]**
- **[LF]**
  The method has linear performance in both time and space.
- **[MK]**
  Assuming that the points are not sampled too densely in certain regions, the running time is linear and, if the point samples are sorted along some coordinate axis, the space is sub-linear.

6. How could the approach be generalized?

- **[AS]** It is not clear how this method could be generalized to higher dimensions. It maybe, however, be generalized to use weighted points (to generate triangulations of adaptive samplings), and also to use points generated from particle systems.
- **[DS]**
  More compact, adaptive, interpolating triangulation could be generated by using weighted generalization of alpha-shapes.
- **[FP]**
  They propose an extension which does ball pivoting on the front edges using an increasing list of radius (rho_0< rho_1<… <rho_n) to improve results in non-uniformed sampled regions. Adaptive sampling could be managed following the generalization to weighted alpha-shapes.
- **[JD]**
- **[LF]**
  This method could possibly be used to reconstruct surfaces sampled with particle systems, or with adaptive sampling techniques (i.e. sample density is increased in high curvature regions perhaps?). The paper mentions extending the method to use a “weighted generalization of alpha-shapes” but at this point I don’t know enough about alpha shapes to understand what this means.
- **[MK]**
  It would be interesting to see if one could extend this approach to computed a weighted alpha shape (in order to account for non-uniform sampling density) using an adaptive data-structure to store points, and still get linear time complexity.

7. If you could ask the authors a question (e.g. “can you clarify” or “have you considered”) about the work, what would it be?

- **[AS]**
- **[DS]**
- **[FP]**
- **[JD]**
- **[LF]**
- **[MK]**