Multi-level Partition of Unity Implicits

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 construct surface models from extremely large point sets using piecewise quadratic functions that capture the local shape of the surface, and then blending them together using multi-level partitions of unity as weighting functions. This is the first method to use partitions of the point set into regions to fit implicit functions to, as opposed to function fitting at every point, in previous methods.
- **[DS]**
  The paper presents a surface representation using multi-level partition of unity (MPU) implicit surfaces. Presents an approach with the 3 key concepts:
  1. Piecewise quadratic functions used as local shape estimates for the surface.
  2. Weighing functions (partition of unity) to blend the local shape functions.
  3. Octree subdivision that adapts based on shape complexity.

  This approach is adaptive based on the desired accuracy.
  The use of implicit functions provide data repairing capabilities for scattered data and an easy way to perform implicit modeling operations.
- **[FP]**
  This paper proposes a method to implicitly define a surface as the zero level of a distance function.
  The space is partitioned in octree cells and a shape function is fitted to each cell. In contrast to previous approaches (such as Hoppe 92), the distance to the surface (i.e., the implicit function evaluation) is not computed respect to a single element of the space partition (in this case, an octree cell) but is a weighted sum. Due to the partition of unity approach and the use of smooth weighting function, the distance function changes smoothly along the 3D space.
- **[JD]**
- **[LF]**
  The paper introduces the partition of unity approach for approximation of the signed distance function of a set of scattered points with normals. The method breaks up the point set using octree subdivision (can handle complex topology). It then uses piecewise quadratic function as local approximations of the surface. These local approximations are blended together using smooth, local weights (can represent sharp features).
- **[MK]**
  This paper proposes an efficient, octree-based, algorithm for reconstructing surfaces from oriented point samples. The approach falls somewhere between global implicit methods and MLS approaches, as the approach partitions space, computes a local implicit surface around each partition, and then glues these together using a partition of unity.
  In addition, the authors also describe how to handle sharp features by clustering points based on their normal orientations, fitting implicits to each of the clusters independently,

and then using the min/max to get a function representing the intersection of the corresponding solids.

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

- **[AS]**
  The key insight of this paper is that multi-level partition of unity weighting functions allow us to reconstruct surface models from very large points sets in very little time.
- **[DS]**
  The use of implicit functions to model the surface by using piecewise-defined shape functions blended with partition of unity weight functions. The local shape estimates are fit for points in each cell of the octree (for approximating surfaces).
- **[FP]**
  Independent and locally accurate signed distance functions are obtained using space refinement (octrees) and polynomial fitting. Blending using PU gives a smooth global distance function.
- **[JD]**
- **[LF]**
  The key insight was perhaps the application of the partition of unity approach to approximation of a signed distance function.
- **[MK]**
  The key idea behind this work is to solve the reconstruction problem locally, and then use a partition of unity to smoothly glue the local solutions together. Using an octree, the reconstruction adapts to the number of points and to the quality of the fit – refining the tree when there are enough points that are not fit well with the current solution.

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

- **[AS]**
  Their use of quadratic functions for distance approximation is not optimal, and other estimations of distance might help accurately reconstruct more complex sharp features.
- **[DS]**
  The input is a point set with normals.
  The sharp feature detection is based on normal clustering which is sensitive to noise.
- **[FP]**
  The main limitation of the method is the requirement of normals. Normals are used to define a global coherent sign of the local distance function.
  Since the method only defines an implicit representation of the function, there are no primitives (triangles, tetrahedron, etc.) from which topological and geometric attributes of the surface can be deduced (on the other hand, the implicit representation easily allows Boolean operations as presented by the authors).
- **[JD]**
- **[LF]**
  The expected input is a point cloud with normals. The method is robust to varying

sample density. A smoothing parameter, alpha, must be specified. Other parameters must be specified to determine how to robustly fit a local shape function to a subdomain.

- **[MK]**
  The method is not well-constrained away from the samples, and hence may produce spurious islands away from the samples when there is noise in the input.

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

- **[AS]**
  Given high enough accuracy, the output is guaranteed to accurately depict the surface, including representation of sharp features.
- **[DS]**
  The output is an implicit function defined at all points in the domain. It'll have a value of zero close to the surface, and positive (inside) or negative (outside) away from the surface. Sharp features will be identified reproduced correctly.
- **[FP]**
  Since the implicit function is a smoothly weighted sum of piecewise quadratic functions, zero level set should be connected but not necessarily smooth (i.e., may have corners or edges). Defining if the zero level set will be always manifold is harder. I think manifold is not guaranteed. For instance, consider points distributed in a cone. The zero level set may still have a non-manifold vertex.
- **[JD]**
- **[LF]**
  The output should be at least piecewise smooth.
- **[MK]**
  Assuming the implicit function does not have zero gradient (which the authors seek to avoid by adding additional constraints at cell corners) the reconstructed surface should be manifold without boundary.

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

- **[AS]**
  The space and time complexity of this approach depends on the complexity of the reconstructed shape rather than the number of data points. This is because higher the complexity of the shape, the deeper the octree, and therefore higher memory requirements, and more cells to fit polynomials to, increasing runtime.
- **[DS]**
  The space/time complexity of this approach depends on the desired accuracy of the representation as well as the complexity of the shape.
  The time complexity is approximately quadratic on alpha (which determines the size of the sphere associated with each octree node).
  Interpolating a surface is more computationally intensive (time and space) than approximating.
- **[FP]**
  As the authors claim, the complexity of the approach depends more on the geometric detail (point distribution) of the input set, than on its size. The authors fix a parameter ϵ,

the level of approximation desired, which guides the octree cell subdision. This parameter ϵ is compared to an input dependent value, which indicates the approximation obtained for the point distribution in a cell neighborhood.
Another parameter associated to the complexity of the approach is α, which indicates the smoothness of the reconstruction. This defines both the weights of samples used to fit the implicit function at each cell, and the weights of local implicit functions in the global implicit functions.
Since each sample point only participates on function fitting of a constant number of cells, the local function fitting phase should be (n). The global implicit function evaluation requires weighting on a constant number of local implicit function, so this is just (1).

- **[JD]**
- **[LF]**
  The method's complexity is dependent on the complexity of the reconstructed shape rather than the number of sample points. The method is faster than FastRBF and can be easily implemented out-of-core given the local nature of PU techniques.
- **[MK]**
  Linear

6. How could the approach be generalized?

- **[AS]**
  This approach could be generalized to an out-of-core implementation due to the local nature of the weighting functions.
- **[DS]**
  A different estimation of the distance function might improve the approach (other than quadratic function). A richer library of local shape approximations can be built to reconstruct sharp features better.
  Well suited for an out-of-core implementation.
  Can be extended to take into account per-point measurement confidences.
- **[FP]**
  The space division is one of the features that can be extended/ adapted from the methodology introduced. In the paper space division is based in a hierarchical octree. Each local implicit function is fitted and has support on a spherical region centered on a cell and with radius proportional to the cell diagonal. Restricting the local functions to be attached to a hierarchical grid (i.e., restricting position and size) is good for performance issues (e.g., finding the relevant local functions active at certain point), but may be is not optimal in approximation quality terms. May be, better results are obtained by adaptively centering and sizing the support of the local functions.
  Another feature susceptible to be generalized is the space of functions used for fitting. This paper use two particular type of functions (polynomial graphs and quadrics) but some other higher order functions may be considered.
- **[JD]**
- **[LF]**
  High order functions could be used to approximate the distance function. The decision structure for alternating between functions could be changed.

- **[MK]**
  Given the more recent work, it seems like clustering normals based on dot-product may be too simple (and sensitive to noise). Is it possible to use a more advanced clustering technique (e.g. bilateral filtering, or the robust technique of Fleishman et al.) to identify distinct point clusters?

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]**
  Have you considered using a k-D tree, instead of an octree, and observing the performance of your algorithm using this data structure?
- **[DS]**
  Which other types of local shape functions would most improve the reconstruction?
- **[FP]**
  The function fitting, for both polynomials and quadrics, seems to be based in traditional least square approach, which usually are not robust to noise and outliers. The results presented in the paper also seem to be obtained from reduced noise set points. How is the performance of the method in the presence of noise?
- **[JD]**
- **[LF]**
  How/why did you choose that set of quadratic functions?
- **[MK]**
  Given the method assumes that normals are given, is it possible to incorporate normal fitting into the local fitting of implicits?