SSD: Smooth Signed Distance 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 an approach for reconstructing surfaces from oriented point sets, where instead of forcing the implicit function to approximate a discontinuous indicator function like in the Poisson Surface Reconstruction approach, it forces the implicit function to be a smooth approximation of the signed distance function to the surface. The gradient of this distance function exists and has approximate unit slope in the neighborhood of the data points, and the vector field to be approximated is constrained to be the gradient of the implicit function. This formulation can be efficiently discretized, and produces high quality adaptive manifold meshes comparable with state-of-the-art algorithms.
- **[DS]**
  Given a set of oriented points, the method reconstructs a watertight surface defined by an implicit equation. The method discretizes the formulation so that the solution reduces to a sparse linear system of equations. The implicit function is taken to be a smooth approximation of the signed distance to the surface. The normal vector data is incorporated into the energy function, so the function does not need to be smoothed. The vector field is constrained to be the gradient of the implicit function. A hybrid FE/FD discretization scheme is presented. Finite element discretization is used for the function values, and a finite differences discretization is used for the gradient and the Hessian.
- **[FP]**
  The authors propose a method to solve for a function (with "the smoothest" gradient field) that implicitly defines the surface. In order to compute this function the authors minimize an energy that is composed by three terms: (1) a data fitting term (to set small value of the implicit function in the sample points), (2) a vector field alignment term (to make the gradient field of the implicit function parallel to sampled normal), and (3) a smoothing gradient field term (by minimizing the norm of hessians). The authors claim that this approach is closely related to Poisson reconstruction (which fails on satisfying condition (3)) but produces sharper results, since it does not requires of smoothing the implicit function. Some results are presented showing sharper reconstructions than Poisson and better performance on filling undersampled regions.
- **[JD]**
  The paper presents another surface reconstruction technique using a signed distance function. The algorithm expands on the popular signed-distance method but uses a finite element/finite differences solution using an octree.
- **[LF]**
- **[MK]**
  The paper proposes a novel approach for computing a smoothed signed distance function from a set of oriented points. To this end, the authors use the fact that the distance function has the properties that on the surface the function has value zero and gradient aligned with the surface normal, while away from the surface the Hessian of the function is approximately zero. This motivates a variational approach to surface reconstruction that combines local (point-wise) value/derivative fitting with global (integrated) bi-Laplacian minimization.

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

- **[AS]**
  The key insight of this paper is the introduction of a regularization term containing the

Hessian in the definition of the energy to be minimized in order to be able to use compactly supported basis functions to represent the implicit function, without worrying about the fact that its behavior away from data points is not well defined. By forcing the square norm of the Hessian to be almost zero, the regularization term makes the gradient of the function nearly constant away from data points, whereas close to the data points, the function approximates the signed distance function. Compact support is good because it results in a sparse matrix in the linear system that needs to be solved for the implicit function.

- **[DS]**
  The approach computes a function that is a smooth approximation of the signed distance to the surface that was a constant gradient far away from the points (minimize the Hessian).
- **[FP]**
  From the set of functions that satisfy data and normal constraints take one with small variation in the gradient field. Discretize and solved the problem efficiently using an octree.
- **[JD]**
  The key insight of the paper is to add to the energy function a regularization term that uses the Hessian. Forcing the square norm of the Hessian matrix to be close to zero makes the gradient of the function almost constant away from the data points, which prevents spurious components appearing far from the data points.
- **[LF]**
- **[MK]**
  The key observation of the paper is that, away from the surface, the smooth signed distance has the property that its gradients are approximately constant, so a reasonable regularization constraint is to minimize the (total) variation of the gradients.

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

- **[AS]**
  One limitation of this method is the assumption that their input point set is oriented.
- **[DS]**
  It doesn't scale well for large datasets.
- **[FP]**
  As in the Poisson case, the first limitation is the requirement of reliable normal.
  In terms of the problem formulation, the authors define the energy function as the sum of three independent terms weighted by certain scalars. Defining the weights of these scalars does not seem to be a trivial task, and it is not clear if they are input dependent or not.
  The authors solve for an optimal solution by discretizing both the gradient and the hessian of the implicit function, and plugging them on the energy function. It is not clear what the loss in accuracy is due to this discretization.
- **[JD]**
  The input is assumed to be oriented. The algorithm can create surfaces where the Poisson, MPU, and D4 wavelets fail.
- **[LF]**
- **[MK]**
  The approach requires solving a bi-Laplacian equation, which is less well-conditioned then solving a Poisson equation, potentially requiring more iterations of an iterative solver.

Near the surface, the signed distance function may exhibit strong variation in the gradient (particularly near regions of high-curvature). As a result, it is possible that as a result the regularization would result in over-smoothing near the surface. (Though this is not manifest in the results.)

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

- **[AS]**
  The output is guaranteed to be a smooth, watertight (crack-free), manifold polygon mesh.
- **[DS]**
  A watertight surface defined by an implicit equation.
- **[FP]**
  Since this is a global implicit function approach the zero level set (thus the reconstructed surface) is expected to be watertight. The meshing algorithm (in this case based in Dual Marching cubes) guarantees manifold conditions.
- **[JD]**
  The output is manifold but not water-tight.
- **[LF]**
- **[MK]**
  The output will be manifold, without boundary (assuming non-vanishing gradients).

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

- **[AS]**
  If the iterative solver is used, then the time complexity of this method is $O(N\sqrt{N})$ .However, if the hierarchical solver is used, then the time complexity is $O(N)$. The space complexity of this method is the space complexity of an octree, $O(N)$.
- **[DS]**
  An octree is used to space complexity is approximately linear.
  Time complexity is more than linear.
- **[FP]**
  As in Poisson case, memory and time of reconstruction of this method seems to be approximately linear in the output resolution (number of triangles). Since this method requires computation of the Hessian norm at each octree cell (whether in closed or discretized form), this may represent significant additional costs compared to Poisson reconstruction.
- **[JD]**
  The authors do not discuss the complexity. The space complexity for storing the matrix is linear because it is sparse.
- **[LF]**
- **[MK]**
  This is difficult to say. The construction of the octree probably has complexity $O(N \log N)$. (Assuming roughly $O(N)$ leaf nodes with points, each point will need to be pushed through $O(\log N)$ nodes to get to the leaf.)
  There is also the question of the efficiency of the solver. This is harder to asses since as the authors provide little information aside from commenting that it is hierarchical. Published methods for solving hierarchically over an octree indicate a complexity between $O(N)$ and $O(N \log N)$. However, running times documented in Screened Poisson Reconstruction indicate a super-linear time that appears closer to $O(N^{1.5})$ – corresponding to the expected efficiency of a non-hierarchical CG solver.

6. How could the approach be generalized?

- **[AS]**
  This approach can be generalized to a streaming or parallel implementation.
- **[DS]**
  Improving scalability.
- **[FP]**
  Since the energy function is constructed as sum of independently weighted terms, there are many adaptations/ extension that can exploit this generality. First, instead of setting scalars weight for the three components, we could set weights per sample (position or normal) or per region (hessian). In such way, we could weight the samples according to the confidence on the registration, or obtain a result that is spatially adaptive (interpolative in some regions, smoother in others).
  A second extension would be a method that evaluates the gradients and hessian of the implicit function in closed form, and then, compute the energy value in closed form too. This seems to be achievable (and not excessively expensive) using compact supported functions such as bsplines as in Poisson.
- **[JD]**
  The algorithm could be generalized to any energy function. The paper's contribution is the Hessian term to the energy function.
- **[LF]**
- **[MK]**
  Hard to say.

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]**
  It is unclear if Hausdorff distance is the right comparison metric for reconstructed shapes. Have you considered any other metric to compare your results against other methods?
- **[DS]**
  What causes the time complexity in your implementation to be more than linear? What ways have you thought of to improve that?
  What benefits does the FE/FD hybrid provide?
- **[FP]**
  1) How do you fit the values of parameters λ0,1and λ2 of the energy function?
  2) What are tradeoffs (in terms of performs and accuracy) between the discretized approach you follow and a closed form computation?
- **[JD]**
  How much quality is lost from the octree approximation of nodes?
- **[LF]**
- **[MK]**
  The mix of FD/FE solvers seems un-necessary. One could encode the whole thing using a finite-elements solver (e.g. using second order B-splines to support the taking of second derivatives). The linear system becomes straightforward to formulate (with matrix entries obtained by integrating products of functions and derivatives) and the Hessian part can be implemented efficiently using the tensor-product structure of the B-splines. As an interesting note, I believe one could obtain the same system if one were to minimize the square of the Laplacian. For the minimization of the Hessian, one gets matrix coefficients:

$$\int_{\mathbb{R}^3} \|H(f)\|_F^2 = \int_{\mathbb{R}^3} \sum_{ij} \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)^2 = \int_{\mathbb{R}^3} \sum_{ij} \frac{\partial^2 f}{\partial x_i \partial x_j} \cdot \frac{\partial^2 f}{\partial x_i \partial x_j} = \int_{\mathbb{R}^3} \left(\sum_{ij} \frac{\partial^4 f}{\partial x_i \partial x_i \partial x_j \partial x_j}\right) \cdot f$$

$$= \int_{\mathbb{R}^3} \left(\sum_{i} \frac{\partial^2}{\partial x_i^2} \left(\sum_{j} \frac{\partial^2 f}{\partial x_j^2}\right)\right) \cdot f = \int_{\mathbb{R}^3} (\Delta^2 f) \cdot f = \int_{\mathbb{R}^3} \Delta f \cdot \Delta f$$