Robust Moving Least-squares Fitting with Sharp Features

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 robust moving least squares (MLS) approach to define a surface from a potentially noisy point set. Using techniques from robust statistics to determine neighborhoods used by MLS, this approach produces a piecewise smooth surface, and is therefore able to preserve sharp features.
- **[DS]**
  The paper introduces a robust moving least-squares technique for reconstructing a piecewise smooth surface from a noisy point cloud. The method introduces the use of a new robust statistics method for outlier detection: the forward-search paradigm.
  The algorithm classifies regions of a point-set into outlier-free smooth regions, which allows for the projection of points onto a locally smooth region. This increases the stability of the projection.
  The algorithm synthesizes new points that reconstruct sharp crease features, which are not part of the input set.
- **[FP]**
  The authors propose a method to do point projection on a piecewise smooth surface. The authors decompose the set of samples in several regions using robust statistical methods (Least Median Squares and Forward Search) and project the point to the closest valid region. The results presented by the authors show reconstructions with sharper edges than global MLS (e.g., PSS) and also seems to be more robust to noise and outliers.
- **[JD]**
  The paper presents a method for point set surfaces that detects sharp features. Previous work smoothed sharp features away with noisy samples. This work allows a surface with sharp edges to be reconstructed from a noisy input point set.
- **[LF]**
  The paper introduces the forward-search paradigm, a robust statistics method for outlier detection, as a technique for dealing with noise and retaining sharp features when computing an MLS-surface.
- **[MK]**
  The paper proposes two new contributions to the area of surface reconstruction. First, it describes a robust forward-search paradigm to generate reconstructions that are stable in the presence of noise. Second, it extends the MLS framework by supporting the reconstruction of surfaces with sharp features.

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

- **[AS]**
  The key insight of the paper is that by using the forward-search paradigm, a new robust statistics method, regions of the point set can be locally classified into multiple outlier-free smooth regions. This allows for points to be projected on a locally smooth region, defining a piecewise smooth surface and increasing the numerical stability of the projection operator.
- **[DS]**
  The key insight of the paper is the introduction of the forward-search paradigm as a new method for outlier detection. This makes the projection operator (onto locally smooth surfaces) more stable and more resistant to noise.

- **[FP]**
  Sharp features reconstruction requires of robust decomposition of the sample set. This decomposition was done by statistical methods able to identify and extend reliable set of samples.
- **[JD]**
  The key insight is to use the forward search algorithm to statistically fit samples to multiple polynomial surfaces. Thus, edges and corners are preserved because points are fitted to one surface and removed from the others.
- **[LF]**
  The key insights of the paper were perhaps application of the forward search paradigm to MLS and the idea that samples of an edge, or sharp feature, may be distinguished from noise by being at the intersection of multiple smooth surfaces.
- **[MK]**
  The key idea behind this approach is, roughly, to support reconstruction of surfaces with sharp features by (locally) decomposing the point set into samples that to belong to different surfaces, reconstructing those surfaces independently, and then using the points on each surface that agree with the other surface definitions.

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

- **[AS]**
  Ambiguity between a noisy smooth region and a sharp feature increases if the sampling density or signal-to-noise ratio are low, and smooth regions may be classified as having a sharp feature or a sharp feature might get smoothed out in the reconstruction. As the two sides of the edges tend towards being co-linear, the reliability of the position of the reconstructed edge decreases. Further, the curve of the edge defined may not be smooth since classification at a point might differ slightly from that at another point.
- **[DS]**
  Noisy data might cause ambiguity between a noisy smooth region and sharp features.
  The reliability of the position of the reconstructed edge decreases as the two sides of the edge tend toward being co-linear.
  The curve of the edge defined by the operator may not be smooth.
- **[FP]**
  The authors suggest that normal information is not required in their approach, but it is not clear if they use normal orientation to validate point projection.
  Due to the locality of the method, dense sampling (especially near sharp features) is a required condition.
  This method does not define a surface decomposition on geometric primitives, neither a characterization of the surface as the zero level of an implicit function. Therefore, it provides limited information on the topological and geometrical attributes of the subjacent surface.
  Finally, the point projection algorithm requires of probabilistic methods (random sampling), which may do not guarantee the stability of the method or the reproducibility of results (I mean, you may require lots of random sampling iterations to get the expected result).
- **[JD]**
  The method assumes oriented points. The authors show that after a certain level of noise edges may be smoothed into a single surface. They have multiple examples of the algorithm's effectiveness on practical data, particularly the drill bit reconstruction which has both sharp and smooth features from a noisy input.

- **[LF]**
  The input is a point cloud and assumed to represent a surface that is smooth over some identifiable region. The user needs provide the location of such a region as a parameter. The method is robust to noise. Like most methods it is vulnerable to undersampling, in this case making it difficult distinguish smooth surfaces from their intersections. This can also happen as intersecting surfaces tend toward being co-linear.
- **[MK]**
  It is not clear that the surface generated in this fashion is in fact connected. In particular, I could not see why it was guaranteed that if one has a small ball of points near the surface (close to a corner) that the image of the points in that ball, under projection, would be a connected 2D patch.
  Also, if I understand correctly the method proposed by the authors is not actually a projection since getting the points to converge to an edge (when all the projections of a point don't agree with some surface) requires an iterative projection approach.
  Like the PSS method, this approach does not explicitly generate a surface.

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

- **[AS]**
  The output surface is piecewise smooth and non-shrinking.
- **[DS]**
  The projection operator defines a piecewise smooth surface from noisy data.
  Handles complex shapes suppressing the shrinking effect (from other plane-fitting and denoising algorithms).
- **[FP]**
  The image of the projection operator (i.e., the set of points that are projection of others) should be smooth due to the use of MLS. Since the surface assignation operator (i.e., the operator that map a point to a particular surface) is discrete, it is not clear if the image of the projection operator is manifold or even connected.
- **[JD]**
  The output is a piece-wise smooth manifold surface. The authors do not discuss whether the surface can have boundaries, but one of their reconstructions shows a boundary due to a limited input.
- **[LF]**
  The output will be piecewise smooth, but is not guaranteed to be smooth at edges.
- **[MK]**
  Not clear

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

- **[AS]**
  Time complexity is $O(n^2)$, and space complexity is also $O(n^2)$.
- **[DS]**
  Unclear.
  Most computationally intensive part seems to be the forward-search and refitting algorithm, which will depend on the number of parameters needed for the model and number of iterations need to get a good estimate.
- **[FP]**
  The first phase of the method requires of identifying the different smooth components in the point neighborhood. This is done by Least Median Squares and Forward Searching. Least Median Squares requires several iterations of random sampling, fitting a model, and residual computation. Each of these iterations in (n).

In each step of Forward Searching it is required a Least Square fitting and global residual evaluation. Therefore, Forward Searching is (ns), where s is the final number of samples in the component. LMS and Forward Searching must be repeated (if required) to get some other components.
The projection stage step use MLS to project on each of the smooth components. Validation and selection of the closest valid projection is (1). If no valid projection exists the projection is iterated.

- **[JD]**
  They note that the algorithm is an order of magnitude slower than the classic MLS algorithm, but they do not discuss complexity.
- **[LF]**
  Complexity was not explicitly mentioned in the paper. The forward-search process makes it slower than classical MLS. Otherwise, the approach is local and so independent of the number of points.
- **[MK]**
  The efficiency of the method is dependent on the speed with which a point's neighbors can be computed. (With a good data-structure, e.g. kD-tree, this should result in a near linear-time algorithm.)

6. How could the approach be generalized?

- **[AS]**
  This approach can be generalized to use priors to improve the faithfulness of the reconstruction, and might also be extended such that the edge curve is guaranteed to be piecewise smooth.
- **[DS]**
  The piecewise smooth surface estimates using the forward-search algorithm can be used to detect sharp features in other surface reconstruction methods, e.g. Ohtake 2003 (which uses normal clustering for edge and corner detection).
  The piecewise operator can be extended such that the edge of the curve is piecewise smooth.
- **[FP]**
  The authors propose a method to segment a set of samples in the neighborhood of a point. This segmentation can be used not just for point projection, but for other applications such as filtering (bilateral filtering).
  The particular statistical methods used by the authors are LMS and Forward Searching, however some other robust techniques may apply to this problem. The local projection operator (MLS) and the space of functions used for approximation (second degree polynomial) are also susceptible to be adjusted.
  Finally, another variant of the method may use a smooth projection (i.e. compute the final projection as a weighted sum of the projections on each component) instead of single component projection. This may improve results in smooth regions of the object, but some care should be taken to not affect sharpness in edges and corners.
- **[JD]**
  The algorithm can be generalized for higher dimensions
- **[LF]**
  The forward-search and local classification approach could be used in any reconstruction method that at some point requires grouping of points with similar normals.
- **[MK]**
  The approach does not explicitly regularize the crease-curves that are generated. In

many applications, it seems like there is a strong prior on the crease curves being smooth 1D curves. Can this be exploited by the algorithm?

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]**
  How is the smoothness in your method defined, since the method does not really define a surface?
- **[DS]**
  Would having normals as part of the input help improve your algorithm or the results?
  Is there a way to do the projection without any use of surface normals?
- **[FP]**
  I will ask the author to clarify surface projection validation. What is the data required and how do they use the data in the validation process.
  They present satisfactory results for objects with sharp features. I would ask them what kind of artifacts the method introduces in smooth objects.
- **[JD]**
  Why did you only use second-order polynomials? Could the approach be generalized to higher order polynomials?
- **[LF]**
  This method could perhaps be used to obtain a relatively smooth, noise free point set. Have you considered any kind of post processing over these points to obtain a smooth, manifold surface?
- **[MK]**
  Can the authors clarify how they determine if the projection onto one surface agrees with the other surface? (How is convexity used? Are they assuming that only normal lines are given, or do they also need to know the normal vectors?)