Robust Moving Least-squares Fitting with Sharp Features

Shachar Fleishman
University of Utah

Daniel Cohen-Or
Tel-Aviv University

Cláudio T. Silva
University of Utah

Abstract

We introduce a robust moving least-squares technique for reconstructing a piecewise smooth surface from a potentially noisy point cloud. We use techniques from robust statistics to guide the creation of the neighborhoods used by the moving least squares (MLS) computation. This leads to a conceptually simple approach that provides a unified framework for not only dealing with noise, but also for enabling the modeling of surfaces with sharp features.

Our technique is based on a new robust statistics method for outlier detection: the forward-search paradigm. Using this powerful technique, we locally classify regions of a point-set to multiple outlier-free smooth regions. This classification allows us to project points on a locally smooth region rather than a surface that is smooth everywhere, thus defining a piecewise smooth surface and increasing the numerical stability of the projection operator. Furthermore, by treating the points across the discontinuities as outliers, we are able to define sharp features. One of the nice features of our approach is that it automatically disregards outliers during the surface-fitting phase.

Keywords: moving least squares, surface reconstruction, robust statistics, forward–search

1 Introduction

Digital scanning devices are capable of acquiring high-resolution 3D models have recently become affordable and commercially available. Modeling detailed 3D shapes by scanning real physical models is becoming more and more commonplace. Current scanners are able to produce large amounts of raw, dense point sets. Consequently, the need for techniques for processing point sets has recently increased. One of the principal challenges faced today is the development of surface reconstruction techniques which deal with the inherent noise of the acquired dataset. When the underlying surface contains sharp features, the requirement of being resilient to noise is especially challenging since noise and sharp features are ambiguous, and most techniques tend to smooth important features or even amplify noisy samples. Moreover, sharp features consist of high frequencies which cannot be properly sampled by the finite resolution of the scanning device in the first place.

Recently, there has been substantial interest in the area of surface reconstruction (or modeling) from point-sampled data. A particularly powerful approach has been the use of the moving least-squares (MLS) technique for modeling point-set surfaces (PSS) [Alexa et al. 2001; Amenta and Kil 2004b; Levin 2003]. One of the main strengths of this approach is the intrinsic capability to handle noisy input, as compared to combinatorial (or topology reconstruction) schemes [Amenta et al. 1998; Bernardini et al. 1999], which rely on clean (or filtered) data. MLS is based on local fitting, and it is naturally framed as an statistical approach to surface reconstruction. Furthermore, the MLS technique makes it easy to compute a very good approximation of the intrinsic properties of the surface such as normal and curvature directly from a noisy point-cloud.

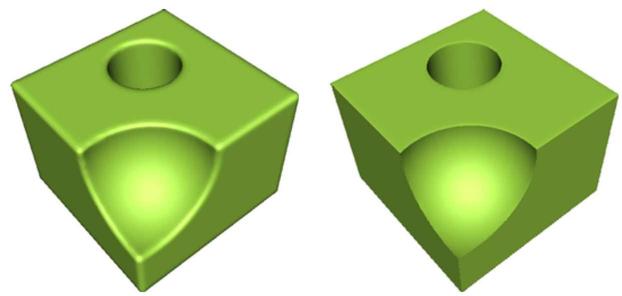


Figure 1: (a) Levin's MLS surface defines a smooth surface. (b) The robust MLS defines a piecewise smooth surface.

In this work, we introduce a piecewise smooth surface definition that is based on Levin's point set surfaces [Levin 2003] by defining a projection operator that accounts for C^1 discontinuities. Our work is related to recent developments in feature-preserving smoothing [Clarenz et al. 2000; Fleishman et al. 2003b; Jones et al. 2003] but unlike smoothing, we define a surface rather than filtering the geometry. Furthermore, since the PSS definition fits a high-order polynomial to the surface, it does not shrink the object.

Points on sharp features are defined by multiple surfaces. Thus, dealing with sharp features requires fitting a number of surfaces locally [Ohtake et al. 2003; Pauly et al. 2003]. This is a non-trivial task since it requires the identification of discontinuities or the locus of the intersection of a number of local smooth surfaces in the presence of noise. If the point data contains reliable normals, they can be used to segment local surfaces [Ohtake et al. 2003]. However, using normals to assist the identification of a discontinuity is a "chicken and egg" problem, since the definition of a normal assumes local smoothness, and its computation in the presence of noise or near a discontinuity is unreliable. Here and in the rest of the paper, by *discontinuities* we refer to the discontinuities in the derivative of a surface.

Our work is based on a powerful, relatively recent robust statistic technique called the forward-search paradigm [Atkinson and Riani 2000; Hadi 1992]. The basic idea in forward search is to start from a small set of robustly chosen samples of the data that excludes outliers. Then to move forward through the data adding observations to the subset while monitoring certain statistical estimates. We use these methods to deal with noise, outliers and sharp features. In our work, sharp features are handled by treating the points across sharp features as outliers. Instead of fitting a single surface locally by the moving least-squares method, we use an iterative refitting algorithm, based on the forward-search algorithm, to classify a neighborhood to multiple local surfaces. Points which are close to more than one local surface are projected on one of its neighboring smooth regions. The local classification along with a new projection procedure defines a piecewise smooth surface where each region is infinitely smooth. Based the robust projection operator, points can be resampled on the piecewise smooth surface by the MLS projection mechanism [Alexa et al. 2001].

The main contribution of this work is a representation for piecewise smooth surfaces, and an algorithm to generate the representation from a noisy data set. We use a new technique from robust statistics that is better suited to detect outliers than previous approaches and thus finds a better fit to a model. In particular, we identify discontinuities in the presence of noise by treating adjacent surfaces and sampling errors as outliers. A unique feature of our reconstruction algorithm is that it synthesizes new points that reconstruct the sharp crease features, which are not part of the input point set.

Before we go on, we would like to note upfront that in the following we will often use the abused term *robustness*. In our work the term robustness does not simply refer to a stable computation but rather to our methodology, which is based on *robust statistics*. In technical terms, we use the term robustness to indicate that we can handle noisy data that is composed of random additive Gaussian noise and possibly a large number of outliers.

2 Background and related work

A main motivation of the type of work described in this paper is the need to model real-world 3D geometry acquired by range scanners. Typically, it is necessary to perform multiple scans to capture the entire geometry, and to register them into a common aligned coordinate system; the output of this process is a raw point set. This point set is noisy and rough, hence the extraction of a thin piecewise smooth surface is a non-trivial task. The whole process is usually exacerbated by the size of the models, which may contain tens of millions of samples or more. Ideally, a surface reconstruction algorithm should be insensitive to noise, and generate a piecewise smooth surfaces which adequately represent the sharp features.

Since the early 1990s, there has been a substantial amount of work in this problem, most noticeably in the general areas of surface reconstruction, point-based modeling, and the use of robust statistics in computer vision. We briefly review the most related works with a view on how they handle noise and whether they are able to model sharp features.

2.1 Moving least-squares surfaces

A point set surfaces [Alexa et al. 2001] is a smooth surface representation of a, possibly noisy, set of points, reconstructed based on a moving least-squares (MLS) technique for surfaces [Levin 2003]. As an approximating scheme, moving least-squares is insensitive to noise. The technique is attractive since the surface is reconstructed by local computations and it generates a surface that is smooth everywhere [Levin 2003]. Recently, different types of PSS formulations have been used for surface reconstruction [Amenta and Kil 2004b; Fleishman et al. 2003a; Mederos et al. 2003; Pauly et al. 2003; Xie et al. 2003].

One appealing feature of point-based representation is the ability to easily make topological changes to an object. Pauly *et al.* [2003] introduced a point-based modeling system that exploits this quality. Based on the PSS definition, they define an implicit function that allows performing constructive solid geometry (CGS) operations on objects. A CSG operation between two objects typically generates a sharp edge. To represent these edges they add new points on the intersection of two given surfaces. In our work, the intersecting surfaces are not given or known a priori, but rather identified and reconstructed in the presence of noise.

Amenta and Kil [2004b; 2004a] study the properties of the surface that is defined by the PSS and the stability of the projection procedure. Their main contribution is in showing the relationship between PSS and extremal surfaces and thus they are able to give an explicit definition of the PSS. Using the extremal surfaces point of view, they define alternative surfaces for surfels. One of their conclusions is that previously defined PSS behaves in an undesirable way when projecting points near edges and corners and thus they suggest new energy functions and perform multiple iterations of the projection until the converge to a surface in order to overcome the problem. Our algorithm handles such sample sets in a more natural way by fitting a piecewise smooth surface to a sample set of a piecewise smooth object rather than fitting a smooth surface to such data. A benefit that we gain from such a definition is increased stability of the projection operator.

2.2 Surface reconstruction

The MLS is only one of many different surface-reconstruction techniques. Pioneering work in surface reconstruction was done by Hoppe *et al.* [1994], who introduce an algorithm that creates a piecewise smooth surface in a multi-phase process that was based on implicit modeling of a distance field. Smoothness is achieved by applying subdivision techniques, and sharp features are defined by two polygons with a crease angle that is higher than a threshold. Another approach is to first reconstruct a mesh [Curless and Levoy 1996; Turk and Levoy 1994] and only then to apply a smoothing process to the mesh that removes noise [Clarenz et al. 2000; Desbrun et al. 1999; Taubin 1995]. The surface normal and viewing direction can be used to consolidate points that were scanned multiple times [Curless and Levoy 1996; Wheeler et al. 1998].

Amenta *et al.* [1998] approached the problem from a computational geometry (combinatorial) viewpoint. Their approach was the first to provided *provable* sampling conditions under which the reconstructed surface was known to be homeomorphic to a smooth compact 3-manifold. Unfortunately, in both theory and practice their technique usually fails on both noisy (and undersampled) models or models with sharp features. Along the same lines (and using geometrical properties of the Delaunay triangulation), Dey and co-workers [to appear] developed techniques that have strong theoretical guarantees for noisy, undersampled, and models with sharp features. Still, since those techniques interpolate the original points, those techniques are intrinsically sensitive to noise in the sense that they will generate a noisy surface out of a smooth model.

An alternative approach is to reconstruct a surface and denoise the input point set in a single unified step. Interpolating a set of points with radial basis functions (RBF) offers a smooth object representation. Typically this requires the minimization of a thin-plate spline energy functional. Computing an RBF interpolation is performed by solving a system of equations of size up to $3N \times 3N$, where N is the number of input points. Carr *et al.* [2001] use a fast solver for RBF that has a complexity of $O(N \log(N))$. Morse *et al.* [2001] use functions with local support, forming a sparse system of equations that can be solved on $O(N^{1.5})$. Since interpolating schemes preserve the noise in the data, Carr *et al.* suggest an approximating variation for an RBF representation of an object. Dinh *et al.* [2001] suggest using a nonsymmetric RBF function aiming at capturing sharp features. They identify edges using covariance analysis of a neighborhood of a point to determine the shape of the function assigned to the point.

Ohtake *et al.* [2003] introduce an implicit function surface representation defined by a blend of locally fitted implicit quadrics (MPU). Each quadric approximates points in a local neighborhood and thus is not sensitive to noise. To reconstruct sharp features they identify edges and corners using normal clustering. If the variation among the normals of a given neighborhood is too large, they cluster the points into sets with similar normals, and fit a quadric to each set separately. The intersection of these quadrics represents the local surface. In our work we follow these ideas and represent sharp features by the intersection of locally fitted surfaces. However, as mentioned in Section 1, the identification of sharp features does not assume the availability of reliable normals, we prefer using robust methods to estimate the locus of local surfaces.

Recent advances in contouring techniques reconstruct sharp features [Ju et al. 2002; Kobbelt et al. 2001; Varadhan et al. 2003]. These methods reconstructs a surface from volumetric data by locally analyzing the vertices of each voxel, assuming noise-free data.

2.3 Robust statistics methods

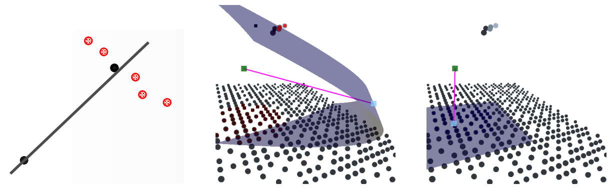
Locally fitting multiple surfaces to points in the area of discontinuity can be regarded as a statistical problem of fitting a model (estimator) to data with outliers. A statistical method is considered to be robust if it has a large breakdown point. Loosely speaking, a breakdown point is defined as the minimal percentage of outliers that can be made arbitrarily large that makes the estimator go to infinity. For example, the breakdown point of the median of a set of values is 50%.

Robust statistics methods have been applied to various computer vision applications [Meer et al. 1991; Stewart 1999]. Sinha and Schunck [1992] introduce a two-stage algorithm for discontinuity-preserving bicubic spline 2.5D surface reconstruction. In the first stage they remove outliers using the least median of squares (LMS) method. In the second stage, they reconstruct the surface using bicubic splines. Miller and Stewart [1996] use ordered statistics to improve the breakdown point of the LMS algorithm. They fit multiple planes to a region by robustly fitting a plane to points in a region, removing the points that have good fit and refitting a second plane. The method we present in this paper is in the same spirit. However, we fit one or more higher order polynomials over 3D data, thus we can reconstruct curved surfaces with no a priori knowledge of the complexity of the reconstructed feature and at the same time we also disregard outliers.

Pauly *et al.* [2004] presented a method for measuring the uncertainty of a point set. They fit a plane to a neighborhood and measure the uncertainty of the neighborhood based on the residuals of the points in that neighborhood. Their method, like other backward methods cannot detect masked outliers (defined below). Backward methods for fitting a model to noisy data work by fitting a model to the entire sample set and then trying to delete bad samples. Unfortunately, as well-known in the statistics literature [Atkinson and Riani 2000], a single or multiple outliers can influence the fitted model in such a way that the good samples are detected as the outliers as demonstrated in Figure 2.

Xie *et al.* [2004] extended the MPU technique to handle noisy datasets. They describe separate procedures for outlier detection and noise removal. For outlier detection they further differentiate between *near* outliers from *far* outliers. For the near ones they employ a thresholding scheme. Far ones are identified by their orientation algorithm. For noise removal, they use an iterative method that defines weights for the points based on how well they fit the surface inside versus outside a user-defined region of space.

Recently Fleishman *et al.* [2003b] and Jones *et al.* [2003] applied the bilateral filter to surface denoising, which can be regarded as a robust statistics technique. For every point, they locally fit a plane to the surface and apply the bilateral filter to the neighborhood of the point. In these works, a single plane is fitted to a point, this plane serves as a parametrization over which the bilateral filter is applied. Paradoxically, a point on a sharp edge that these algorithms aim to preserve resides on two planes rather than one. In this paper, we identify these two planes and synthesize the surface as the intersection of these surfaces.



(a) (b) (c)

Figure 2: A single outlier can cause a least squares fit to fail. In (a) we show a set of points in 2D and the least squares fit to these points (black). A backward method identifies the outliers with respect to the initial guess, thus points in red will be erroneously deleted. Levin's projection can make the analogous erroneous fit (b). Our projection ignores outliers and thus produces the expected result (c).

3 Robust estimation

In our work we deal with fitting a surface to a set of points in 3D. Generally, in statistics, regression deals with fitting, or estimating a model from a *sample set*. The classic method for fitting a model to data is linear regression using least-squares. However, a single sample with a large error, an *outlier*, can change the fitted model arbitrarily. Robust estimation techniques try to fit a model to data that may contain outliers. In this brief review, we concentrate on methods that are most relevant to us, For more details see [Huber 1981]; some applications for computer vision are described by Black and Sapiro [1999].

In this work, we use the *forward search* method for outlier identification [Atkinson and Riani 2000]. This method can deal with multiple outliers, as well as *masked outliers*. Masked outliers are outliers that cannot be identified from the statistics of a model that is fitted to the entire sample set, that is the masked outliers influence the fitted model in such a way that they are cannot be recognized as the source of the misfit. Figures 2(a) and (b) shows an example where a single outlier causes a least squares fit to produce the undesirable result, from the figure (see caption) it is clear that any attempt to identify the outliers based on that fit will fail.

3.1 Least median of squares

The least medians of squares (LMS) is a robust regression method that estimates the parameters of the model β by minimizing the median of the absolute *residuals*, these are defined as the difference between the measurement and estimation: for the ith sample $r_i = f(\mathbf{x_i}) - y_i$. That is, we search the parameters β that minimizes the median of the residuals:

$$\underset{\beta}{\operatorname{argmin}} \underset{i}{\operatorname{median}} |f_\beta(\mathbf{x}_i) - y_i|, \tag{1}$$

and thus can reliably fit a model to data that contains up to 50% outliers.

Equation (1) can be solved using the following random sampling algorithm; k points are selected at random, and a model is fitted to the points. Then the median $r_{i,\beta}$ of the residuals of the remaining $N-k$ points is computed. The process is repeated T times to generate T candidate models. The model with minimal $r_{i,\beta}$ is selected as the final model. If g is the probability of selecting a single good sample at random from our sample set, then the number of iterations that are required to have a probability of success of P can be computed by $P = 1 - (1 - g^k)^T$ [Rousseeuw and Leroy 1987]. A small value of k does not use all of the available sample to fit a model, while a larger value of k requires more iterations. If the value of k is too large, the algorithm becomes sensitive to noise.

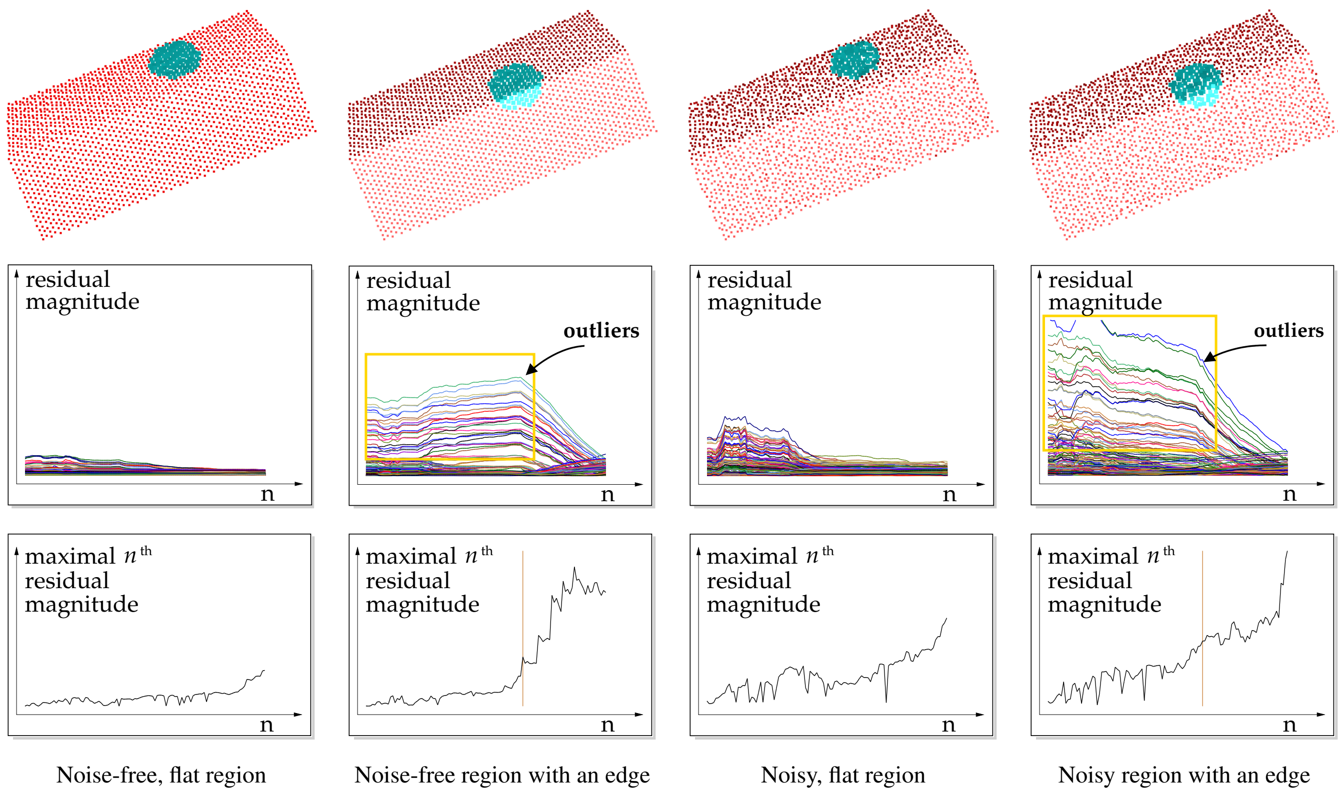


Figure 3: Monitoring the forward search. The first row shows the input model with the neighborhood that is examined. The second row shows the residual plot; each curve in these plots shows the behavior of the residual of a single sample as more and more samples are included in the fitting. The X-axis is the number of samples that are used for fitting, and the Y-axis is the magnitude of the residual. Observe that for the neighborhoods that contain an edge, there are a number of points with large residuals and all of them have a sharp drop in the magnitude of their residual at some point . This is the point where samples from the "wrong" side of the edge enter the sample set. This is precisely the point that we are interested in finding. The third row shows plots of the ith maximal residuals, where the Y-axis is the value of the maximal residual. We use a threshold on the maximal residual (marked in orange) to automatically find the time where outliers enter the sample-set.

3.2 Forward search and iterative refitting

The forward search algorithm [Atkinson and Riani 2000; Hadi 1992] is a robust method that avoids the need to fix k. It begins with a small outlier-free subset and then iteratively refines the model by adding one sample at a time. The initial model is computed using the LMS algorithm using a small k value, typically $k = p$ for a model with p parameters. Then at every iteration i, the i samples with lowest residuals are used to estimate the parameters of the model. During the forward search a number of parameters can be monitored to detect the influential points. Typically, the forward search will add the good-samples first and only when these are exhausted, outliers will be added. Riani and Atkinson [Atkinson and Riani 2000] suggest several statistics to be monitored. For their purposes, these are plotted on a graph and inspected visually. They suggest to monitor the residual-plot (Figure 3 middle), The ith Studentized residual, Cook's distance or modified Cook distance.

The *residual plot* (Figure 3) is a standard method in regression analysis for identifying outliers. The plot contains the residuals of all of the samples on the Y axis and the X axis is the time-step, in which new samples are added. The residual plot is then examined manually to determine the time when outliers started to enter the sample set and thus the classification to good samples and outliers is a post-process. The power of the forward search can be seen when observing the shape of the residual plot: in the example shown in Figure 3, samples from one side of the edge are used and the residuals on the left-hand side of the plot can be roughly divided into two groups, one that has low residuals and are packed at the bottom, and the other has high residuals and are scattered above. When outliers begin to enter the sample set, there is a clear indication in the residuals plot, the residuals of the outliers decrease and the residuals of the good samples increase. The time that this occurs is marked by an orange line in the maximal residuals plot in Figure 3. In our technique we monitor the maximal residual (Figure 3), and we show a method for setting the threshold of the maximal residual (the exact same orange line) so that the process is automated. During the process of the forward search, typically a single sample is added at each iteration, thus sorting the samples according the confidence of them being good samples; samples that enter first to the sample-set have higher confidence.

Iterative refitting. To fit a model to a sample set $\mathcal{S}$ that was sampled from multiple processes, we use an iterative refitting algorithm: First, we fit a model to a subset $\mathcal{S}_1$ of the data, using the forward search algorithm and identify the rest of the data as outliers. Next, we remove the samples that were fitted $\mathcal{S} = \mathcal{S} \backslash \mathcal{S}_1$, and repeat the process until $\mathcal{S}$ is empty. Figure 4 shows an example of this process in two dimensions.

4 Local classification and projection

We present an algorithm that gets as input a set of points $\mathcal{S}$ and classifies this set to a number of subsets, each corresponding to a smooth region of the surface (Figures 4 and 6). This set of points is the neighborhood of some input point $\mathbf{x}$. We adapt the above

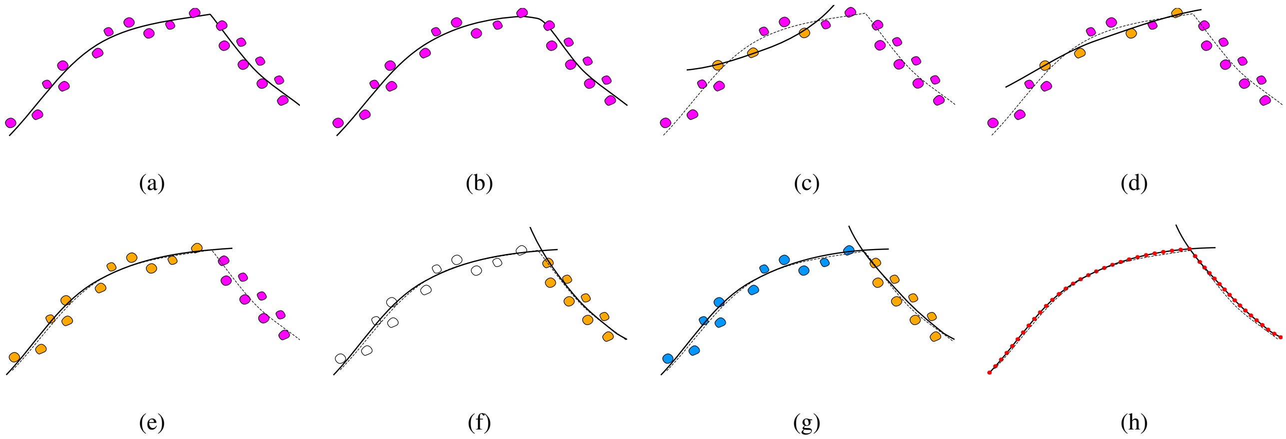


Figure 4: The input noisy data can be interpreted as piecewise smooth surface (a) or as a smooth surface as in (b). We identify the sharp features with the iterative refitting procedure. First, we robustly fit a surface to a small subset of the points in (c). Next, we incrementally add points with smallest residual and refit the surface to the updated subset (d). The final fit of the forward search is shown in (e). The remaining points are regarded as outliers to the first surface, these are used again to refit another surface (f). The result is a surface that is defined as the intersection of the two surfaces in (g). Finally the piecewise smooth surface is reconstructed by resampling (h).

forward search algorithm to classify the point set, i.e. partition $\mathcal{S}$ to subsets each of which is an outlier-free sample of a smooth regions. Next, we present a projection operator which projects a given point near the surface onto the point set surface as illustrated in Figure 5. The latter extends the basic moving least-squares projection operator [Levin 2003] by using the classification to deal with sharp features. The new projection operator defines a piecewise smooth surface, rather than a surface that is smooth everywhere.

Given a sample point $\mathbf{x}$ and its neighborhood $\mathcal{S}$. Our goal is to locally fit a number of polynomials to points in $\mathcal{S}$. The number of polynomials is equal to the number of smooth regions that are in the neighborhood of $\mathbf{x}$. A single polynomial is fitted to points that lie on a smooth region, and multiple polynomials for points that are near an edge.

4.1 Initial robust estimator

As described in Section 3.1, the LMS algorithm randomly selects a number of points, fits a model, and computes the median of residuals. The LMS, as a statistical method, assumes that the samples (points) are independent, and requires a large number of iterations to achieve a high probability of finding a good estimator. We take advantage of the geometric prior that the points sample a surface to significantly accelerate the process and improve its stability in this geometric setting. That is to say that we iterate over all the points in $\mathcal{S}$ and fit a surface to the small neighborhood of each point, following the assumption that neighborhoods sample a single surface.

To fit a polynomial to a set of points in 3D space, it must reside over a parameter domain. We define the parametric domain of a point using eigenvector analysis of the points in its neighborhood Q. Then a polynomial is fitted to the points in Q and the median of residuals of the points in $\mathcal{S}$ is computed. Since we expect that more than a single surface fits $\mathcal{S}$, instead of using the median, we use a kth ordered-statistics to grade a polynomial [Miller and Stewart 1996]. This simply means that the residuals are sorted and we examine the value of the kth residual.

4.2 Forward search on point sets

To find a subset of points that lie on a smooth region, we apply the forward search algorithm, fitting a bivariate polynomial of degree two. First we compute a robust estimator for a small number of points using the algorithm described above. The result is a reference plane and an initial set of points Q.

The second step of the forward search is to iteratively add one point to the set Q at each iteration. At the ith iteration, we use the i points with the lowest residuals, until the largest residual $r_i > r_t$, where r_t is the threshold of maximal tolerated residual. Again we use the geometric prior by setting the candidate points for Q_i to be the points in Q_{i-1} and their immediate neighbors. Figure 4(c–e) illustrates the process.

The threshold of maximal tolerated residual is globally computed for the entire model. Following the mechanism suggested by Fleishman *et al.* [2003b], the user marks a small region on the surface that is known to be smooth, the system fits a polynomial to that region and measures the largest residual to set the value of r_t. Following is a pseudocode of the forward search algorithm:

```
[Estimator, Q] = Fwd(PointSet S) {
  [ParameterDomain, Q] = LMS(S, p)
  for (i = p+1; i < size(S); ++i) {
    Candidates = (Q ∪ ImmediateNeighbors(Q)) ∩ S
    Estimator = LeastSquareFit(ParameterDomain, Q)
    if (r_i > r_t) break;
    R = Residuals(Estimator, Candidates)
    Sort(R) and Candidates respectively
    // Q holds the i points with smallest residual
    Q = Candidates(1...i)
  }
}
```

4.3 Defining the piecewise smooth surface

The moving least-squares projection operator [Levin 2003] assumes that the surface is smooth everywhere, thus it reconstructs a smooth surface as in Figure 1(a). Using the classification process described above, we define a projection operator that projects a point onto the piecewise smooth surface. This classification allows us to project points on a locally smooth region rather than a surface that is smooth everywhere, thus defining a piecewise smooth surface. Furthermore, by treating the points across the discontinuities as outliers, we are able to define sharp features.

Given a point $\mathbf{x}$, we first analyze its neighborhood. If the neighborhood is determined to be smooth, we project it using Levin's

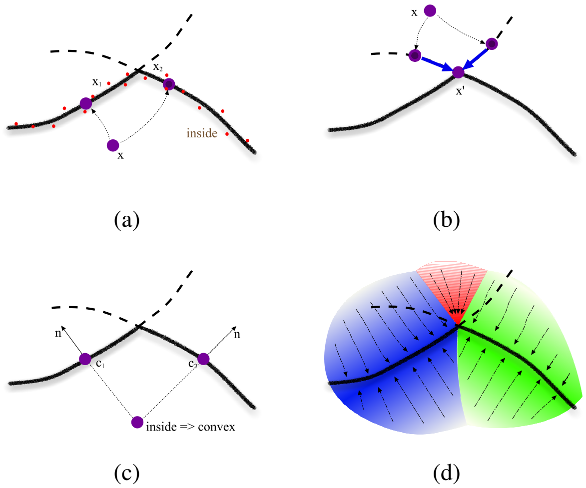


Figure 5: Projection on the surface near an edge is performed by projecting the input point $\mathbf{x}$ on the two surfaces, verifying that is on the surface and choosing $\mathbf{x}_1$, the point that is closest to $\mathbf{x}$. Some points do not project to a valid point on the surface as in (b), thus we project them to the edge. Determining whether a region is convex or concave is done by checking the position of the point that is closest to the two lines $(\mathbf{c}_1, \mathbf{n}_1)$ and $(\mathbf{c}_2, \mathbf{n}_2)$ with respect to the normals $\mathbf{n}_1, \mathbf{n}_2$ (c). (d) shows different regions of space and where points from those regions project to.

method. Otherwise we classify the neighborhood of the point to subsets of smooth regions of the surface and discard outliers. Then we project the point on the closest smooth region of the surface.

Determining whether a neighborhood is smooth or not, is performed by locally fitting a polynomial to the neighborhood and measuring the maximal residual of the points in the neighborhood. If the maximal residual is smaller than the threshold r_t as defined in Section 4.2 then the neighborhood is defined as smooth.

If the neighborhood is not smooth, we use a subset of the neighborhood that is sampled from a smooth region of the surface. We apply the procedure that was described in Section 4.2 to obtain a subset of the neighborhood that samples a smooth region. Then we apply the iterative refitting algorithm to obtain the rest of the smooth regions of the neighborhood. The result is a classification of the neighborhood to one or more subsets as in Figure 4(g). In the case that there is only one subset, the rest of the points are outliers and we simply discard them and project the point as if it were a smooth region.

When we identify two or more subsets, the surface is defined as the intersection of the smooth surfaces defined by the different neighborhoods. Our new projection works by first projecting $\mathbf{x}$ on one surface and then use the other surfaces to check if the point belongs to the intersection or not. In case the two points belong to the intersection, we pick the one that is closest to $\mathbf{x}$ as shown in Figure 5(a). In case none of the points belong to the intersection as in Figure 5(b), we iterate on this process until we converge to the edge as Pauly *et al.* [2003]. Figure 5(d) shows the regions of space and their target projection. Note that unlike Pauly *et al.*, we are not strictly seeking for a point on the intersection of two surfaces, but rather a point on the surface that may or may not be on the intersection curve of the two surfaces.

To check if a point that was projected on one surface agrees with the other surface, we first determine if the neighborhoods form a convex or concave region and then use this information to make inside/outside tests. To determine if the region is convex or concave, we compute the normal to the centroid of each subsets. The normals are oriented based on the input approximated normal; these

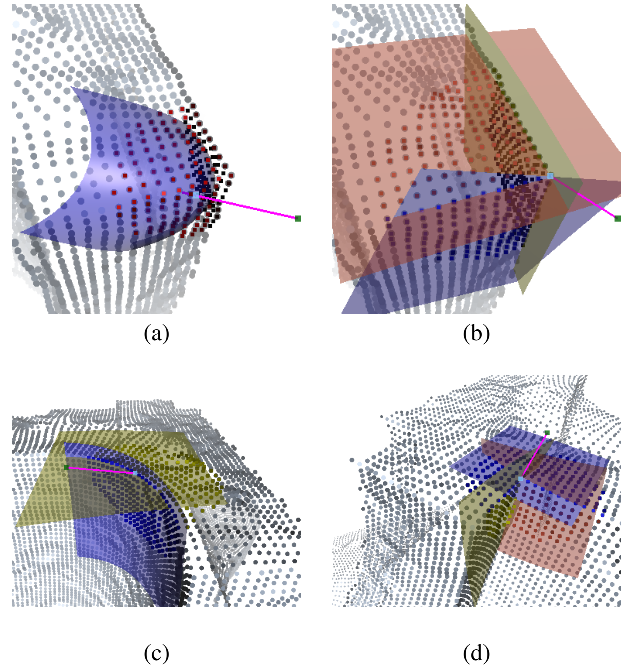


Figure 6: Examples of classified regions and projections. (a) shows Levin's projection of a point near a corner. (b) shows the three surfaces near the corner that were identified by the algorithm and a projection on the surface that is defined by them. (c) and (d) shows similar results for an edge with curved region and a different type of corner.

can be the vector from the point toward the scanner or any other approximation that maintains the consistency of orientation. Then we find the point that is closest to the two rays that are defined by these representative points and their normals (Figure 5(c)). If that point is inside the object then the region is convex, otherwise it is concave. Note that for corners where three surfaces meet there are more involved cases, yet the same principle applies. In Figure 6 we show three examples of classification to regions and projection of a point using those classifications.

5 Results

We have implemented the technique presented in the previous section and tested it on a large number of different scenarios. We report here on three particular ones: a clean synthetic model, a synthetic model with added random noise and the raw output of two different scanners: a Cyberware™ and 3rdtech Deltasphere™ scanners. The new projection procedure is an order of magnitude slower than the classic MLS projection procedure. Clearly any application that uses the robust projection should separate the classification part of the algorithm from the projection part of the algorithm and thus amortize the cost of the classification.

Figure 1(a) shows a reconstructed surface by Levin's projection operator. Naturally, in this reconstruction the edges are smoothed out. In Figure 1(b), we show the surface that is reconstructed by the new projection operator.

To test the ability of the procedure to handle both features and noise, we have added uniformly distributed random noise to the *fandisk* model. The noise is uniformly distributed in the range of 0.2% of the bounding box of the model. In Figure 7 we show the noisy input model on top, and the reconstructed model in the middle and bottom of the figure.

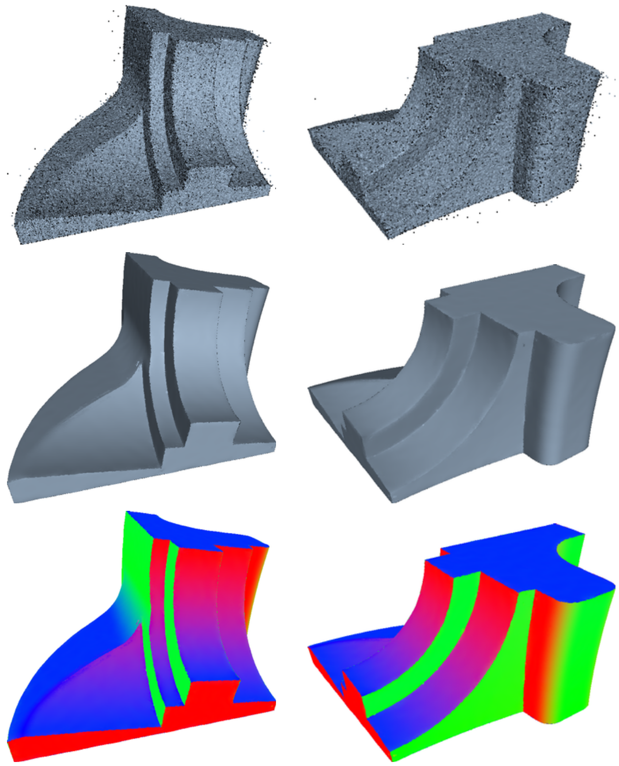


Figure 7: Reconstruction of the *fandisk* model. On the top is the fandisk model corrupted with random noise of 0.2% of the bounding box of the object. Normals to the points were computed using an eigenvector analysis of the covariance matrix using eight nearest neighbors. In the middle is the same model that was resampled with the our method. On the bottom is the same reconstructed model color-coded with the normals to the points.

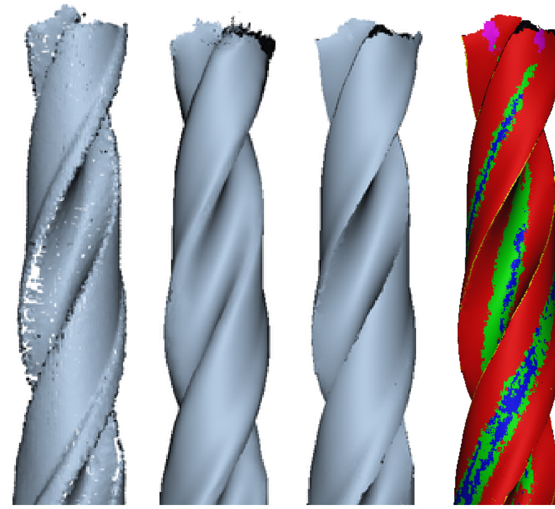
In Figure 8 we show the reconstruction of a drill that is scanned by a Cyberware$^{\text{TM}}$ scanner. The complex geometry of the model leaves some undersampled regions and generates outliers as can be seen in Figure 8(a). We compare Levin's reconstruction in Figure 8(b) to the robust projection (c). In (d) we color-coded the object by the number of surfaces that are in the neighborhood of the point, except for the points that were projected to the edge that are colored with yellow. An interesting thing to note about the color-codes is the points that are colored with green; these were identified as single smooth region with some outliers; in this example, most of the green points are inside a smooth neighborhood, but close to an edge. This happens since the points on the other side of the edge were identified as outliers.

In Figure 9 we show a resampling of an object that is the raw output from a Deltasphere$^{\text{TM}}$ scanner. The scanned model in extremely noisy and contains outliers, as can be seen on (a) and (d). We resampled the object and show the result as a smooth rendering, comparing Levin's reconstruction and the robust reconstruction in (b) and (c) respectively. In (d) we superimposed the resampled model over the input model to show the non-shrinking thin reconstruction of the noisy model.

We define a sharp feature at the intersection of multiple surfaces, thus we can reconstruct an edge where some of the data is missing as shown in Figure 10. This is a unique property of our technique compared to other surface reconstruction techniques.

Implementation and parameters. In our implementation we handle regions that have up to three surfaces that meet at a single point such as corners of a cube (e.g, Figure 6(b)). The parameters are a threshold on the largest allowed residual that is set as described in Section 4.2 and the minimal neighborhood size for a surface L_s. From that we set the neighborhood size of a point to be $N_s = 6L_s$, since a point near a corner that is at the intersection of two concave surfaces and a convex surface (as in Figure 6(d)) has approximately three times more points on the convex region, we need five times L_s and we add a few more points as to compensate for uneven sampling and outliers. The kth ordered statistics (Section 4.1) for the initial robust estimator is also set to L_s.

The forward search algorithm as described in Section 4.2 adds a single sample at each iteration and solves a least squares system at each iteration. In the tests that are presented here, we allow adding multiple points at each iteration as long as their residual is within the allowed tolerance and the maximal number of points that we add is not more than 40% of the current size of Q.



(a) (b) (c) (d)

Figure 8: Reconstruction of a drill scan by a Cyberware$^{\text{TM}}$ scanner. (a) the input point-set; note the missing data near and the roughness of the data. (b) reconstruction using Levin's projection. (c) reconstruction using the robust MLS. (d) points are colored by the number of surfaces that lie near them: blue for smooth regions, green for a single surface with outliers, red for two surfaces, magenta for three surfaces and yellow for points that are projected to the edge.

Limitations. Noisy data is always prone to ambiguity between a noisy smooth region and a sharp feature. The presented algorithm declares a smooth region as smooth whenever a polynomial of degree two can be tightly fitted to the local neighborhood. If the sampling density or the signal-to-noise ratio are too low, we may classify smooth regions as containing a sharp feature and vice versa as shown in Figure 11. Furthermore, the reliability of the position of the reconstructed edge decreases as the two sides of the edge tend toward being co-linear.

The presented projection operator defines a piecewise smooth surface, however the curve of the edge that is defined by the operator may not be smooth since the classification at one point may differ slightly from one point to another. Extending the projection operator such that the edge of the curve will be piecewise smooth as well is among our future goals.

6 Summary and conclusions

We have presented a method to locally classify smooth regions in point-sampled objects, using this classification, we have pre-

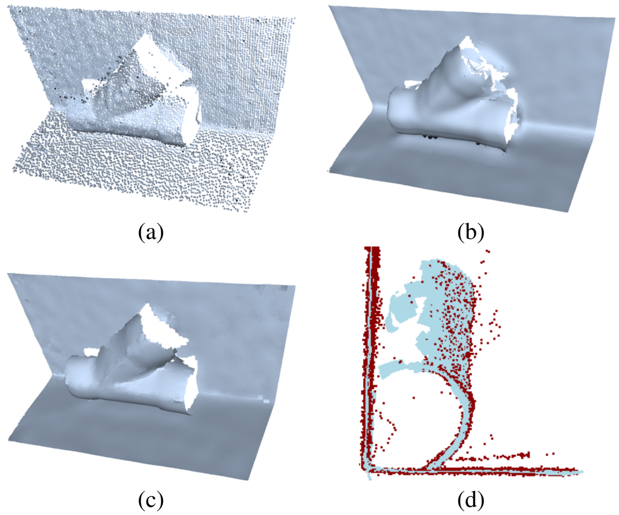


Figure 9: A reconstruction from a raw Deltasphere™ scan of a pipe. (a) input data. (b) a reconstruction using Levin's projection. (c) a reconstruction using the robust projection. (d) The reconstructed surface (blue) is superimposed over the input data (red); note the thin reconstruction from the noisy data.

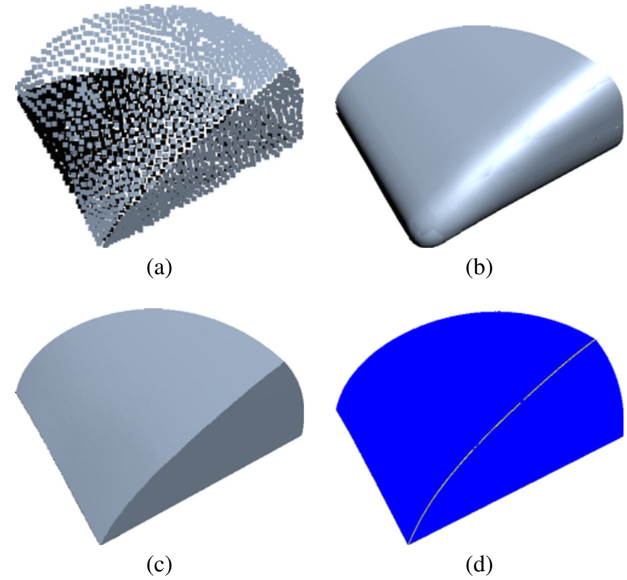


Figure 10: Reconstruction of missing data: in (a) we show an input point sample of a corner of an object, where samples near the edge are missing. (b) is a reconstruction using Levin's projection. (c) is a reconstruction using the robust MLS. (d) points that were projected to the edge are marked in yellow.

sented a projection operator that defines a piecewise smooth surface. With the projection operator we have presented applications for resampling and reconstructing piecewise smooth surfaces from noisy data. We believe that the forward search and the local classification will have numerous other applications. For example, the classification can be used as a basis for grouping points in the MPU [Ohtake et al. 2003] method.

The method uses tools from robust statistics to operate well in the presence of noise, identifies outliers and ignore them. The main tool that we use is the forward-search algorithm which has a significant advantage in detecting outliers over commonly used "backward" methods. Our use of non-planar estimator allows us to handle complex shapes and suppress the shrinking effect that is inherent in plane-fitting based surface fitting and denoising algorithms. Our approach to deal with sharp features is based on the simple observation that a sharp point is defined by more than a single local surface. The classification to local smooth neighborhoods leads to moving least squares operator that considers points only from smooth neighborhoods, while avoiding samples across sharp features. Amenta and Kil [2004b] observed that the point-set surfaces projection operator may be unstable near sharp features. The local classification to smooth regions presented in this work improves the stability of the projection operator near sharp features and outliers.

In our experiments we found that the second degree polynomials is effective in the lack of a prior. However, if some prior is given, the faithfulness of the reconstruction can significantly increase. In some applications, one has higher-level priors; for example it might be known in advance that the scanned surface consists of a conic section, or that it has a circular boundary. Improving the robustness with given priors is a topic for future work.

Acknowledgements

This work is partially supported by the Department of Energy under the VIEWS program and the MICS office, the National Science Foundation under grants CCF-0401498, EIA-0323604, and OISE-0405402, a University of Utah Seed Grant, the Israel Science Foundation (founded by the Israel Academy of Sciences and Humanities), and the Israeli Ministry of Science. The drill model is courtesy of Sergei Azernikov (Technion - Israel Institute of Technology). We thank Elaine Cohen and Ross Whitaker for use of the Deltasphere™ scanner.

References

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. In *IEEE Visualization 2001*, 21–28.

AMENTA, N., AND KIL, Y. J. 2004. The domain of a point set surface. In *Eurographics Symposium on Point-Based Graphics*, 139–147.

AMENTA, N., AND KIL, Y. J. 2004. Defining point-set surfaces. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004) 23*, 3, 264–270.

AMENTA, N., BERN, M., AND KAMVYSSELIS, M. 1998. A new voronoi-based surface reconstruction algorithm. In *ACM SIGGRAPH 1998*, 415–422.

ATKINSON, A. C., AND RIANI, M. 2000. *Robust Diagnostic Regression Analysis*. Springer.

BERNARDINI, F., MITTLEMAN, J., RUSHMEIER, H., SILVA, C., AND TAUBIN, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 5*, 4, 349–359.

BLACK, M. J., AND SAPIRO, G. 1999. Edges as outliers: Anisotropic smoothing using local image statistics. In *2nd Conf. on Scale-Space Theories in Computer Vision*, 259–270.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and representation of 3d objects with radial basis functions. In *ACM SIGGRAPH 2001*, 67–76.

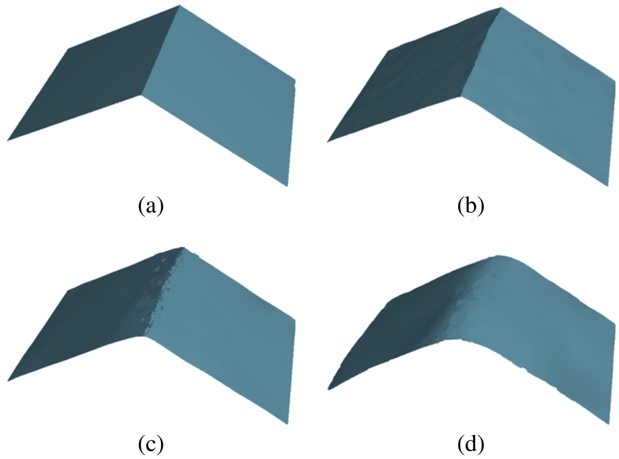


Figure 11: Reconstructions with increasing noise level. (a) Clean wedge data. (b) Successful reconstruction with added noise of 1% of the diagonal of the bounding box. (c) As the magnitude of the noise increases to 2%, the algorithm misclassify some regions. (d) With a noise level of 10%, the edge is completely smoothed out.

CLARENZ, U., DIEWALD, U., AND RUMPF, M. 2000. Anisotropic geometric diffusion in surface processing. In *IEEE Visualization 2000*, 397–405.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *ACM SIGGRAPH 1996*, 303–312.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *ACM SIGGRAPH 1999*, 317–324.

DEY, T. to appear. *CAMS/DIMACS volume on Computer-Aided Design and Manufacturing*. American Mathematical Society, ch. Sample Based Geometric Modeling.

DINH, H. Q., TURK, G., AND SLABAUGH, G. 2001. Reconstructing surfaces using anisotropic basis functions. In *ICCV 2001*, 606–613.

FLEISHMAN, S., ALEXA, M., COHEN-OR, D., AND SILVA, C. T. 2003. Progressive point set surfaces. *ACM Transactions on Graphics 22*, 4, 997–1011.

FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral mesh denoising. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) 22*, 3, 950–953.

HADI, A. S. 1992. Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society: Series B 54*, 3, 761–771.

HOPPE, H., DEROSE, T., DUCHAMP, T., HALSTEAD, M., JIN, H., MCDONALD, J., SCHWEITZER, J., AND STUETZLE, W. 1994. Piecewise smooth surface reconstruction. In *ACM SIGGRAPH 1994*, 295–302.

HUBER, P. J. 1981. *Robust Statistics*. Wiley, New York (NY).

JONES, T. R., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics 22*, 3, 943–949.

JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. 2002. Dual contouring of hermite data. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002)*, 339–346.

KOBBELT, L. P., BOTSCH, M., SCHWANECKE, U., AND SEIDEL, H.-P. 2001. Feature sensitive surface extraction from volume data. In *ACM SIGGRAPH 2001*, 57–66.

LEVIN, D. 2003. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*. Spinger-Verlag, 37–49.

MEDEROS, B., VELHO, L., AND FIGUEIREDO, L. 2003. Moving least squares multiresolution surface approximation. In *XVI Brazilian Symposium on Computer Graphics and Image*, 19–26.

MEER, P., MINTZ, D., KIM, D. Y., AND ROSENFELD, A. 1991. Robust regression methods for computer vision: A review. *International Journal of Computer Vision 6*, 1, 59–70.

MILLER, J. V., AND STEWART, C. V. 1996. Muse: Robust surface fitting using unbiased scale estimates. In *CVPR '96*, 300.

MORSE, B. S., YOO, T. S., RHEINGANS, P., CHEN, D. T., AND SUBRAMANIAN, K. R. 2001. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Proceedings of the International Conference on Shape Modeling and Applications*, 89–98.

OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. 2003. Multi-level partition of unity implicits. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) 22*, 3, 463–470.

PAULY, M., KEISER, R., KOBBELT, L. P., AND GROSS, M. 2003. Shape modeling with point-sampled geometry. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) 22*, 3, 641–650.

PAULY, M., MITRA, N., AND GUIBAS, L. 2004. Uncertainty and variability in point cloud surface data. In *Eurographics Symposium on Point-Based Graphics*, 77–84.

ROUSSEEUW, P., AND LEROY, A. 1987. *Robust Diagnostic Regression Analysis*. John Wiley & Sons.

SINHA, S. S., AND SCHUNCK, B. G. 1992. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 1, 36–55.

STEWART, C. V. 1999. Robust parameter estimation in computer vision. *SIAM Review 41*, 3, 513–537.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *ACM SIGGRAPH 1995*, 351–358.

TURK, G., AND LEVOY, M. 1994. Zippered polygon meshes from range images. In *ACM SIGGRAPH 1994*, 311–318.

VARADHAN, G., KRISHNAN, S., KIM, Y. J., AND MANOCHA, D. 2003. Feature-sensitive subdivision and iso-surface reconstruction. In *IEEE Visualization 2003*, 99–106.

WHEELER, M. D., SATO, Y., AND IKEUCHI, K. 1998. Consensus surfaces for modeling 3D objects from multiple range images. In *ICCV '98*, 917–924.

XIE, H., WANG, J., HUA, J., QIN, H., AND KAUFMAN, A. 2003. Piecewise g^1 continuous surface reconstruction from noisy point cloud via local implicit quadric regression. In *IEEE Visualization 2003*, 259–266.

XIE, H., MCDONNELL, K. T., AND QIN, H. 2004. Surface reconstruction of noisy and defective data sets. In *IEEE Visualization 2004*, 259–266.