Carnegie Mellon University
Research Showcase @ CMU

Robotics Institute | School of Computer Science

2007

Denoising Manifold and Non-Manifold Point Clouds

Ranjith Unnikrishnan
Carnegie Mellon University

Martial Hebert
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/robotics

Part of the Robotics Commons

Published In
18th British Machine Vision Conference (BMVC).

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Robotics Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Denoising Manifold and Non-Manifold Point Clouds

Ranjith Unnikrishnan Martial Hebert
Carnegie Mellon University, Pittsburgh, PA 15213
ranjith,hebert@cs.cmu.edu

Abstract

The faithful reconstruction of 3-D models from irregular and noisy point samples is a task central to many applications of computer vision and graphics. We present an approach to denoising that naturally handles intersections of manifolds, thus preserving high-frequency details without oversmoothing. This is accomplished through the use of a modified locally weighted regression algorithm that models a neighborhood of points as an implicit product of linear subspaces. By posing the problem as one of energy minimization subject to constraints on the coefficients of a higher order polynomial, we can also incorporate anisotropic error models appropriate for data acquired with a range sensor. We demonstrate the effectiveness of our approach through some preliminary results in denoising synthetic data in 2-D and 3-D domains.*

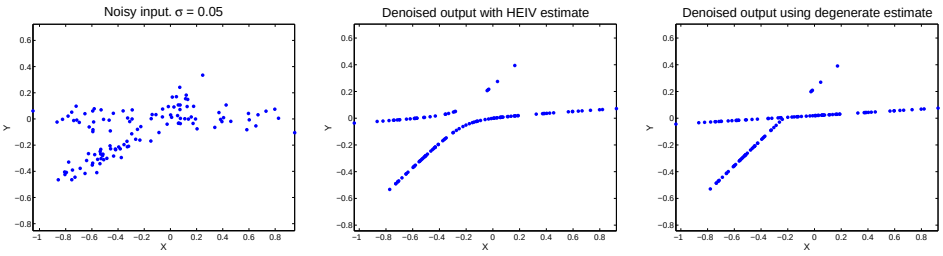
1 Introduction

Surface reconstruction from unorganized point samples is a challenging problem relevant to several applications, such as the digitization of architectural sites for creating virtual environments, reverse-engineering of CAD models from probed positions, remote sensing and geospatial analysis. Improvements in scanner technology have made it possible to acquire dense sets of points, and have fueled the need for algorithms that are robust to noise inherent in the sampling process.

In several domains, particularly those involving man-made objects, the underlying geometry consists of surfaces that are only piece-wise smooth. Such objects possess sharp features such as corners and edges which are created when these smooth surfaces intersect. The reconstruction of these sharp features is particularly challenging as noise and sharp features are inherently ambiguous, and physical limitations in scanner resolution prevent proper sampling of such high-frequency features.

This paper proposes a denoising technique to accurately reconstruct intersections of manifolds from irregular point samples. The technique can correctly account for the anisotropic nature of sensing errors in the sampled data under the assumption that a noise model for the sensor used to acquire the points is available. The method does not assume prior availability of connectivity information, and avoids computing surface normals or meshes at intermediate steps.

* Prepared through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-209912.



(a) (b) (c)

Figure 1: Example of denoising a toy dataset by global fitting of an implicit degenerate polynomial (a) Input data consisting of points from two intersecting line segments corrupted with uniform Gaussian noise of std. deviation $\sigma = 0.5$ (b) Denoised data using an implicit quadratic fit with the HEIV estimator [6]. Note that the sharp feature formed by the intersection is not preserved. (c) Denoised output after imposing degeneracy constraints on fit coefficients fixes this problem.

1.1 Related Work

There have been several proposed approaches to recover geometry from noisy point samples. They may be coarsely categorized as based on computational geometry, local regression, or implicit function fitting.

In general, past approaches have often made simplifying assumptions about the data due to the ill-posed nature of the problem. (1) Methods based on classical regression typically assume that the geometry can be treated locally as a smooth surface, which is clearly a problem at surface intersections. (2) Most approaches assume the noise in the data to be isotropic and homogeneous, perhaps because they often lead to convenient closed-form analytical expressions. However, noise is almost always highly directional and dependent on the distance of the point to the sensor. This is, for example, the case with laser range scanners. Ignoring the anisotropy in the noise model typically results in a systematic bias in the surface reconstruction [6]. (3) Some methods assume the reliable availability of additional information about the geometry, such as connectivity information (meshes) and surfaces normals, and try to produce estimates of geometry that agree with this information. However, the estimation of both these quantities is error-prone. Estimation of differential quantities like surface normals and tangents is difficult in the presence of noise even for relatively smooth surfaces [7, 10], and is of course not even well-defined at intersections.

Several methods based on *computational geometry* have been developed and rigorously analyzed in the literature [3]. Many algorithms in this category come with theoretical guarantees of accuracy in the reconstruction but their applicability is largely restricted to dense low-noise datasets.

Surface estimation from noisy point samples may be posed naturally as an instance of the local *regression* problem from classical statistics. A popular non-parametric technique in this category is locally weighted regression, also known in its more general form as Savitzky-Golay filtering. As explained in [4], it adapts well to non-uniformly sampled data and exhibits less bias at boundaries. The moving least squares (MLS) technique [5] builds on this by first computing a locally approximating hyperplane and then applying a locally weighted regression procedure to the data projected to the hyperplane. The tech-

nique works well with noise but is unable to reproduce sharp features due to its implicit assumption of a single locally smooth surface.

Fleishman *et al.* [8] fit quadratic polynomials locally to data and used standard techniques from robust statistics in the fitting process. The technique relied on an initially finding low-noise local regions to obtain a reliable estimate of the quadratic fit, which may not always be feasible.

Wang *et al.* [13] proposed a more complicated procedure involving a sequence of voxelization and gap-filling, topological thinning and mesh-generation. Based on local connectivity, each voxel is classified as being at a junction, boundary and surface interior. The procedure has several points of failure, particularly at regions that are not densely sampled with respect to the voxel size.

The method presented in this paper combines the strengths of some of the previous approaches. We modify a locally weighted smoother to implicitly represent potentially multiple linear subspaces through a degenerate high-order polynomial. This allows us to explicitly model edge intersections instead of trying to fit a highly non-smooth surface. The use of a local smoother preserves the adaptability to varying sample density. By posing the regression as a constrained energy minimization problem, we can easily incorporate anisotropic error models in the data. We outline the algorithm in Section 2 and examine its behavior through several experiments in Section 3.

2 Constrained Local Regression

In this section, we describe a modified regression algorithm that will enable us to recover noise-free surfaces from noisy point cloud data, while preserving high-frequency features in the geometry. We will first consider the case of 2-D data to simplify the explanation of the main idea.

2.1 Problem definition and approach

We assume that we are given a set of points $\{\mathbf{x}_i\} \in \mathbb{R}^d$ that are assumed to be noisy observations of the positions of true points $\{\hat{\mathbf{x}}_i\} \in \mathbb{R}^d$ that lie on a locally continuous, but not necessarily smooth surface. The associated noise covariances $\Lambda_i \in \mathbf{S}_+^d$ at each point are assumed to be known, for instance, through a noise model of the sensor used to acquire the points. The points are assumed to be *irregular*, in the sense that they do not follow a known regular sampling distribution, and *unstructured* in the sense that the local connectivity of the points, such as in the form of a mesh, is not available.

Our goal will be to compute the true position $\hat{\mathbf{x}}_i$ corresponding to each observed point $\mathbf{x}_i$. The operating assumption will be that points in a local neighborhood, $\mathcal{N}(\mathbf{x}_i)$ of $\mathbf{x}_i$ may be modeled as belonging to one or more linear subspaces. This naturally suggests a maximum likelihood (or equivalently defined minimum energy) formulation of the problem, subject to the constraint that the noise-free points lie on one or more subspaces. Since the parameters of the models, number of models, as well as the association of the points to each subspaces are unknown, a popular strategy is to attempt a procedure of iterative model fitting and data association, such as Expectation-Maximization (EM). However, such iterative procedures tend to be error prone when performed with few and noisy data points, as may be expected for our problem.

Instead, we propose to model the problem as one of maximum likelihood subject to *two* types of constraints. The first type of constraint ensures that each noise-free point in

the neighborhood of interest lies on a high-order polynomial whose degree is an upper bound on the number of subspaces in that neighborhood. The second type of constraint is a function of the coefficients of the polynomial, which restricts the family of allowable polynomials to degenerate forms that can represent combinations of linear subspaces. In practice, we will sometimes relax the constraint of degeneracy to make the optimization problem more tractable at the expense of admitting a single non-linear manifold but restrict them to locally developable surfaces.

2.2 Constraints in the 2-D case

In the case of 2-D data, each local neighborhood can be modeled as as consisting of a pair of linear subspaces. Thus locally the shape may be described implicitly as a zero-level set of the equation $(\boldsymbol{\gamma}_1^{\mathrm{T}}\mathbf{x}+d_1)(\boldsymbol{\gamma}_2^{\mathrm{T}}\mathbf{x}+d_2)=0$, where $\boldsymbol{\gamma}_i \in \mathbb{R}^2$, $d_i \in \mathbb{R}$ are the parameters for each of the two linear subspaces (lines in the case of 2-D data). Note that this subsumes the case where the subspaces coincide. Expanding out the terms yields an inhomogeneous 2nd degree polynomial in 2 variables, which we will refer to as x and y corresponding to each spatial dimension.

Let us denote the coefficients of each monomial in the polynomial as given by the expression

$$\theta_1 x^2 + \theta_2 y^2 + \theta_3 xy + \theta_4 x + \theta_5 y + \theta_6 = 0. \tag{1}$$

This may be rewritten in matrix form as

$$\begin{bmatrix}\mathbf{x} & 1\end{bmatrix}\begin{bmatrix}2\theta_1 & \theta_3 & \theta_4\\ \theta_3 & 2\theta_2 & \theta_5\\ \theta_4 & \theta_5 & 2\theta_6\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix} = \begin{bmatrix}\mathbf{x} & 1\end{bmatrix} A \begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix} = 0. \tag{2}$$

It is a known result in algebraic geometry that a quadratic in two variables reduces to a product of two linear factors only if A is singular [1]. In fact, the case where A has only rank one corresponds to the case where the subspaces (lines) coincide.

The determinant in this case may be written explicitly to yield the equality

$$4\theta_2\theta_2\theta_6 + \theta_3\theta_4\theta_5 - (\theta_2\theta_4^2 + \theta_1\theta_5^2 + \theta_6\theta_3^2) = 0, \tag{3}$$

which can be used to constrain the solution for the θ_i's. We will denote such constraints on the coefficients of the polynomial as $\boldsymbol{\phi}(\boldsymbol{\theta}) = 0$.

2.3 Constrained optimization

Together with the constraint on coefficients we can pose the task as a constrained optimization problem defined at each point of interest $\mathbf{x} \in \{\mathbf{x}_i\}$ given by

$$\min \sum_i w_i(\mathbf{x})(\mathbf{x}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}}\Lambda_i^{-1}(\mathbf{x}_i - \hat{\mathbf{x}}_i), \tag{4}$$

subject to two sets of constraints. The first set of constraints is $\boldsymbol{\theta}^{\mathrm{T}}\mathbf{v}(\hat{\mathbf{x}}_i) = 0\ \forall i$ where $\boldsymbol{\theta} \in \mathbb{R}^m$ is the vector of monomial coefficients and $\mathbf{v}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^m$ is the mapping from the d-dimensional point to the monomials formed by its coordinates. For the 2-D case ($d = 2$), the number of monomial terms $m = 6$. The second constraint is that on the monomial coefficients, which is (3) in the case of 2-D data.

The weighting term $w_i(\mathbf{x})$ is used to give more importance to points closer to the point of interest $\mathbf{x}$. We can define $w_i(\mathbf{x})$ using a kernel loss function, such as a truncated Gaussian function centered at $\mathbf{x}$, so as to suitably delineate the neighborhood of interest $\mathscr{N}(\mathbf{x})$. Our implementation uses the Epanechnikov kernel $w_i(\mathbf{x}) = 1 - \|\mathbf{x}-\mathbf{x}_i\|^2/\sigma^2$ for $\|\mathbf{x}-\mathbf{x}_i\| < \sigma$ and 0 elsewhere, chosen because of its finite support and asymptotically optimal properties in related tasks such as kernel regression [12]. Here σ determines the length scale, which may be chosen differently for each $\mathbf{x}$. We comment on its selection later in Section 2.5. In what follows, we will sometimes drop the dependence on $\mathbf{x}$ in the notation for clarity, with the understanding that the optimization problem is being solved for points in a local neighborhood of *each* $\mathbf{x} \in \{\mathbf{x}_i\}$.

The standard approach to solving such a constrained optimization problem is by first forming the Lagrangian

$$\sum_i \tfrac{1}{2} w_i (\mathbf{x}_i - \hat{\mathbf{x}}_i)^{\mathrm{T}} \Lambda_i^{-1} (\mathbf{x}_i - \hat{\mathbf{x}}_i) + \sum_i \lambda_i \boldsymbol{\theta}^{\mathrm{T}} \mathbf{v}(\hat{\mathbf{x}}_i) + \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{\theta}), \tag{5}$$

where $\{\lambda_i\}$ and $\boldsymbol{\alpha}$ are the Lagrange multipliers.

To proceed further, we linearize the equations around the current estimate of $\mathbf{x}_i$'s and $\boldsymbol{\theta}$. Let $\Delta\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{x}_i$ and $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_0 - \boldsymbol{\theta}$, where $\boldsymbol{\theta}_0$ is the current estimate of the true $\boldsymbol{\theta}$. To reduce notational clutter, we denote $\nabla\mathbf{v}(\mathbf{x}_i)$ by $\nabla\mathbf{v}_i$ and $\nabla\boldsymbol{\phi}(\boldsymbol{\theta}_0)$ by $\nabla\boldsymbol{\phi}_0$. This yields the equation

$$\tfrac{1}{2}\sum_i w_i \Delta\mathbf{x}_i^{\mathrm{T}} \Lambda^{-1} \Delta\mathbf{x}_i \;\; + \sum_i \lambda_i \left(\boldsymbol{\theta}_0^{\mathrm{T}} \mathbf{v}(\mathbf{x}_i) + \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} \Delta\boldsymbol{\theta} + \boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i \Delta\mathbf{x}_i\right) \;\; + \boldsymbol{\alpha}^{\mathrm{T}} \left(\boldsymbol{\phi}(\boldsymbol{\theta}_0) + \nabla\boldsymbol{\phi}_0 \Delta\boldsymbol{\theta}\right) = 0. \tag{6}$$

Taking derivatives with respect to $\Delta\boldsymbol{\theta}$, $\Delta\mathbf{x}_i$ and the Lagrange multipliers yields the system of equations:

$$w_i \Lambda_i^{-1} \Delta\mathbf{x}_i + \lambda_i \boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i = 0 \qquad\qquad \sum_i \lambda_i \mathbf{v}^{\mathrm{T}}(\mathbf{x}_i) + \boldsymbol{\alpha}^{\mathrm{T}} \nabla\boldsymbol{\phi}_0 = 0 \tag{7}$$

$$\boldsymbol{\theta}_0^{\mathrm{T}} \mathbf{v}(\mathbf{x}_i) + \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} \Delta\boldsymbol{\theta} + \boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i \Delta\mathbf{x}_i = 0 \qquad\qquad \boldsymbol{\phi}(\boldsymbol{\theta}_0) + \nabla\boldsymbol{\phi}_0 \Delta\boldsymbol{\theta} = 0. \tag{8}$$

The solution to the above set of equations can be written as

$$\Delta\boldsymbol{\theta} = -\boldsymbol{\phi}(\boldsymbol{\theta}_0)(\nabla\boldsymbol{\phi}_0^{\mathrm{T}} \nabla\boldsymbol{\phi}_0)^{-1} \nabla\boldsymbol{\phi}_0 \tag{9}$$

$$\lambda_i = w_i (\boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i^{\mathrm{T}} \Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0)^{-1} \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} (\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}) \tag{10}$$

$$\Delta\mathbf{x}_i = -\tfrac{1}{w_i} \Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0 \lambda_i = -\Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0 (\boldsymbol{\theta}_0^{\mathrm{T}} \nabla\mathbf{v}_i^{\mathrm{T}} \Lambda_i \nabla\mathbf{v}_i \boldsymbol{\theta}_0)^{-1} \mathbf{v}(\mathbf{x}_i)^{\mathrm{T}} (\boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}). \tag{11}$$

The above solutions to the linearized constrained optimization problem suggests an iterative technique in which a candidate initial value of $\boldsymbol{\theta}_0$ is computed and the values of $\boldsymbol{\theta}$ and the $\hat{\mathbf{x}}_i$'s are progressively modified until the constraints are satisfied. The initial value of $\boldsymbol{\theta}_0$ may be chosen as the result of an unconstrained optimization using the Fundamental Numerical Scheme (FNS) algorithm [2] or the related Heteroscedastic Errors in Variables (HEIV) method [6] based on solving a generalized eigenvalue problem.

Related formulations: At this point, we wish to comment on some related work to clarify some superficial similarities. The use of a high-order polynomial product to represent a combination of (low-order polynomial) subspaces is not new. Work by Taubin [9] fit complex 3-D curves to data, and used a high-order polynomial to represent the intersection of surfaces that formed the curve. It used an approximation to the distance function

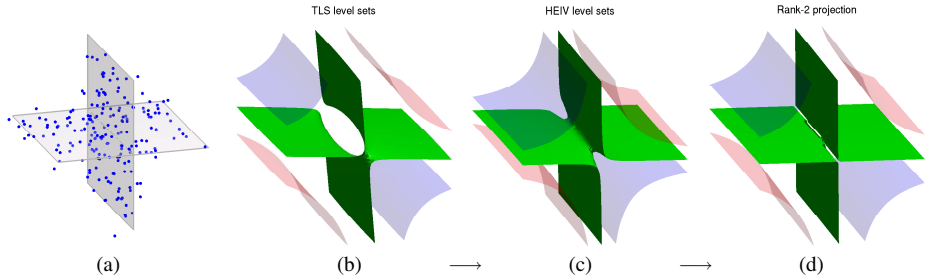


Figure 2: Illustration of sequence of optimization steps in an example of global fitting of (a) noisy observations of points lying on two planes. Level-set surfaces are shown at values 0 (green), 0.15 (red) and -0.15 (blue), and are drawn for parameters estimated with (b) TLS, which are used to initialize solution to the (c) HEIV estimate, which when subject to degeneracy constraints yields the best fit at the intersection of the planes as shown in (d).

that reduced the fitting problem to an easily solvable generalized eigenvector problem, but implicitly made the assumption of *uniform* noise covariance on the points. Vidal *et al.* [11] proposed the Generalized Principal Components Analysis (GPCA) algorithm to model combinations of linear subspaces. However, they did not consider noise in the points, and have to resort to a separate estimation procedure to compute the parameters of the individual subspaces.

In contrast, the formulation in this section explicitly incorporates a heteroscedastic noise model on the points. We use a separate constraint to capture the desired degeneracy of the polynomial as part of the optimization procedure, instead of resorting to post-processing of the result. Lastly, our focus is on *local* rather than global fitting of the data, since the data in our application cannot necessary be described globally by linear subspaces.

2.4 Constraints in the 3-D case

In the case of 3-D data, we consider the choice of model corresponding to an upper bound of 2 linear subspaces (planes) in each local neighborhood under consideration. This may be described formally as a zero-level set of the equation $(\boldsymbol{\gamma}_1^\top \mathbf{x} + d_1)(\boldsymbol{\gamma}_2^\top \mathbf{x} + d_2) = 0$, where $\mathbf{x} \in \mathbb{R}^3$ and $\boldsymbol{\gamma}_i \in \mathbb{R}^3$, $d_i \in \mathbb{R}$ are the parameters for each of the two planes. Note that this again subsumes the case where the subspaces coincide. Expanding out the terms yields an inhomogeneous 2nd degree polynomial in 3 variables (denoted x, y and z).

Let us denote the coefficients of each monomial in the polynomial as given by the expression

$$\theta_1 x^2 + \theta_2 y^2 + \theta_3 z^2 + \theta_4 xy + \theta_5 yz + \theta_6 xz + \theta_7 x + \theta_8 y + \theta_9 z + \theta_{10} = 0. \tag{13}$$

This may be rewritten in matrix form as

$$\begin{bmatrix} \mathbf{x} & 1 \end{bmatrix} \begin{bmatrix} 2\theta_1 & \theta_4 & \theta_6 & \theta_7 \\ \theta_4 & 2\theta_2 & \theta_5 & \theta_8 \\ \theta_6 & \theta_5 & 2\theta_3 & \theta_9 \\ \theta_7 & \theta_8 & \theta_9 & 2\theta_{10} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x} & 1 \end{bmatrix} A \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = 0. \tag{14}$$

Algorithm 1: DenoiseByConstrainedFitting($\{\mathbf{x}_i\}$, $\{\Lambda_i\}$)

Data: Points $X = \{\mathbf{x}_i\} \in \mathbb{R}^d$ with noise covariance $\{\Lambda_i\}$

begin

  for $\mathbf{x} \in X$ **do**

    Compute **weights** $w_i = k(\mathbf{x} - \mathbf{x}_i)$ where k is a loss function such a Gaussian

    Find the **total least squares solution** $\boldsymbol{\theta}_{\text{TLS}}$ to the unconstrained fitting problem. The least square solution is simply equal to the minimal eigenvector of the weighted covariance matrix formed by the $\mathbf{v}(\mathbf{x}_i)$'s, i.e. the minimal eigenvector of $\sum_i w_i \mathbf{v}(\mathbf{x}_i)\mathbf{v}(\mathbf{x}_i)^{\mathrm{T}}$

    Use $\boldsymbol{\theta}_{\text{TLS}}$ to initialize the iterative solution to an **unconstrained optimization** procedure [6]. The solution to the unconstrained problem $\boldsymbol{\theta}_{\text{HEIV}}$ can be obtained through an fixed-point iteration procedure given by:

$$S(\boldsymbol{\theta}(k))\boldsymbol{\theta}(k+1) = \lambda_k C(\boldsymbol{\theta}(k))\boldsymbol{\theta}(k+1) \tag{12}$$

    where λ_k is the smallest generalized eigenvalue, and S and C are given by:

$$S(\boldsymbol{\theta}) = \sum_i \frac{A_i}{\boldsymbol{\theta}^{\mathrm{T}} B_i \boldsymbol{\theta}} \qquad C(\boldsymbol{\theta}) = \sum_i B_i \frac{\boldsymbol{\theta}^{\mathrm{T}} A_i \boldsymbol{\theta}}{(\boldsymbol{\theta}^{\mathrm{T}} B_i \boldsymbol{\theta})^2}$$

    with $A_i = w_i \mathbf{v}(\mathbf{x}_i)\mathbf{v}(\mathbf{x}_i)^{\mathrm{T}}$ and $B_i = \nabla \mathbf{v}_i^{\mathrm{T}} \Lambda_i \nabla \mathbf{v}_i$

    Iteratively enforce the **degeneracy constraint** (3) using equations (9) and (11) (or (14) and (15) in the case of 3-D) along with the unit norm constraint $\|\boldsymbol{\theta}| = 1$ and initializing with $\boldsymbol{\theta}_{\text{HEIV}}$

  end

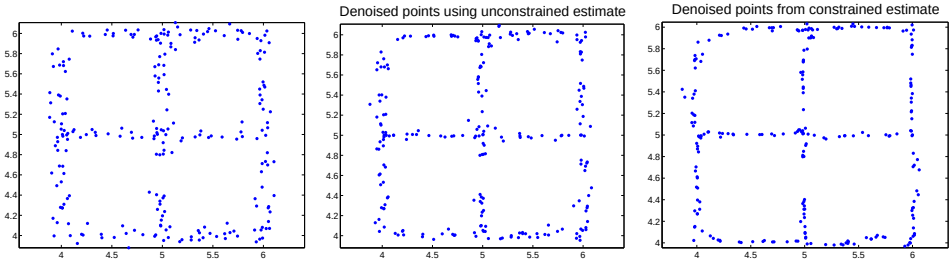
end

Following the argument in Section 2.2, it is easy to see that matrix A must be of rank 2 for the associated quadric surface to represent a pair of planes. This is equivalent to the constraints that the determinant of A as well as each of its 3×3 minors are zero. We have observed it sufficient to relax the constraints on the minors and retain the constraints only on the principal minor formed by the degree 2 coefficients, as

$$\det(B) = \det\left(\begin{bmatrix} 2\theta_1 & \theta_4 & \theta_6 \\ \theta_4 & 2\theta_2 & \theta_5 \\ \theta_6 & \theta_5 & 2\theta_3 \end{bmatrix}\right) = 0. \tag{15}$$

Geometrically, the use of this particular subset of constraints restricts the family of surfaces represented by the polynomial coefficients to the family of parallel or intersecting planes, and cylinders. Using the parameters estimated with this subset of constraints, we may then construct the matrix A, find its rank-2 approximation using its SVD decomposition, and recover the parameters of the degenerate polynomial from the rank-2 matrix.

Figure 2 illustrates the sequence of steps involved in estimating the polynomial coefficients for a synthetic dataset consists of noisy points lying on two planes intersecting at right angles. Level-set surfaces are displayed for the polynomial coefficients estimated at each step of fitting all the points. It can be seen that the TLS solution misfits the geometry, the HEIV solution tends to oversmooth the intersection (as in Figure 1 for 2-D data) and enforcing the degeneracy constraints recovers the true geometry in this example.



(a) (b) (c)

Figure 3: Example of denoising a toy dataset by *local* fitting of an implicit degenerate polynomial (a) Input data consisting of points from six line segments corrupted with spherical Gaussian noise of std. deviation $\sigma = 0.5$ (b) (b) Denoised data using an implicit quadratic fit with the HEIV estimator [6]. (c) Denoised output after imposing degeneracy constraints on coefficients.

2.5 Algorithm and Implementation

From the solution of the constrained optimization problem in the previous section, we may construct our denoising procedure as given in Algorithm 1. We draw attention to some details that influence the performance of the proposed method.

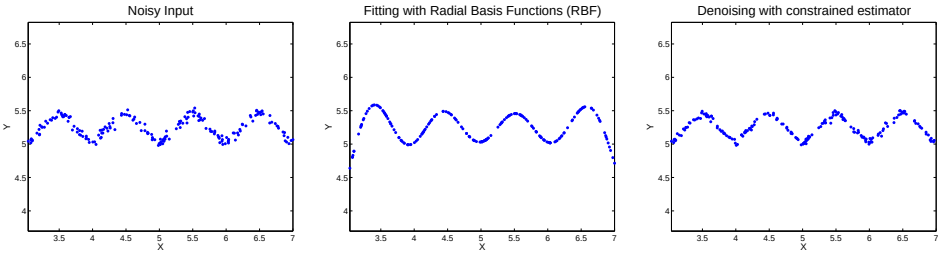
Support radius: The choice of support radius used to compute the weights w_i in the kernel function has a significant influence on the algorithm in two ways. First, the proposed method assumes an upper bound of 2 subspaces in the volume of interest, which need not be the case for any choice of support size. The chosen support radius must be one for which the modeling assumption is valid, conditional on there always existing such a choice. Secondly, even when the assumption of number of subspaces is valid, there is a tradeoff between choosing too small a radius, risking poor estimates due to the fewer number of points, or too large a radius, risking the unfavorable influence of points that do not belong to the local model.

We currently use a heuristic strategy of choosing the support radius that gives the best fit, in a maximum likelihood sense, to the corresponding neighborhood of the interest point, excluding the point itself to prevent the trivial solution of zero radius. In practice, we have observed that when the number of manifolds is under- or over-estimated, this strategy tends to reduce the support radius and show bias toward a one-manifold solution when enforcing the degeneracy constraint. However, this is an area in need of further study.

Robustness: The use of weights w_i also suggests the use of robust statistics to identify outliers to the model [8]. One strategy to identify points that have a large influence on the estimated model parameters, such as using eigenvector perturbation bounds [10] for the generalized eigenvalue problem (12) or using influence functions. In our experiments, we use a simple greedy strategy of evaluating leave-one-out fitting score and ignoring the point as an outlier if it is not a good fit with its neighbors.

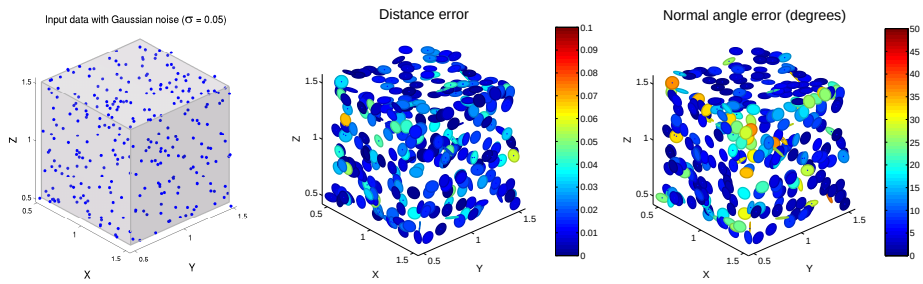
3 Experiments

We performed a series of controlled experiments of synthetic data in known configurations to evaluate the behavior of the denoising algorithm. Figure 3 shows an example where



(a) (b) (c)

Figure 4: Example of denoising samples from a triangular wave function (a) Input data corrupted with spherical Gaussian noise of std. deviation $\sigma = 0.5$ (b) Denoised data using radial basis function based smoother with Gaussian kernel. (c) Denoised output after *local* degenerate polynomial fitting.



(a) (b) (c)

Figure 5: Example of denoising samples from 3 faces of a regular cube (a) Input data corrupted with uniform noise of std. deviation $\sigma = 0.02$. Denoised points are shown with patches color coded by (b) distance error and (c) surface normal angle error.

an Epanechnikov loss function with bandwidth 0.3 was used to denoise a 2x2 square grid pattern of points. The use of a constraint enforcing degeneracy in the polynomial can be seen to preserve the intersections better than using an HEIV smoother.

Figure 4 compares the proposed fitting procedure with a standard interpolation algorithm based on radial basis functions (RBF). The RBF algorithm has two parameters [14]. The first controls the width of the Gaussian kernel which influences the locality of the smoothing. The other controls the tolerance to fitting error, i.e. a value of zero would lead to interpolation between the points, while higher values allow greater fitting error. The parameters were tuned so that the results best matched the ground-truth in the sense of least-square error. It can be seen that the proposed algorithm does a better job of preserving sharp changes in the function and is more stable at the boundary, while the RBF function tends smooths over the high curvature regions.

In Figure 5, we test the proposed algorithm on 300 noisy 3-D samples (spherical Gaussian with std. dev. 0.05) from 3 faces of a unit cube, and compared it against using the HEIV estimator from [6]. The use of the proposed estimator reduced the minimum error in normal angle over the dataset from 0.67° to 0.46° and the median distance of the points to their corresponding planes from 0.012 to 0.009 units.

4 Conclusions

In this document, we investigated the strategy of fitting local degenerate high-order polynomials to data to more faithfully represent and estimate high-frequency variations in point-sampled surfaces. The proposed strategy helps to address the inherent inability to perform differential analysis at non-manifold regions, such as intersections of curves, without actually having to estimate the parameters of component manifolds.

A current open problem is the judicious selection of the support region of the loss function. Too small a value results in a fragmented reconstruction, while the use of too large a value degrades the solution due to the influence of outliers to the implicit model. Work on an analytical solution to the optimal support radius to replace our current heuristic is in progress.

References

[1] K. Barrett. Degenerate polynomial forms. *Communications in numerical methods in engineering*, 15(5), 1999.

[2] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. FNS, CFNS and HEIV: A unifying approach. *Journal of Mathematical Imaging and Vision*, 2005.

[3] T. K. Dey. *Curve and Surface Reconstruction*. Cambridge University Press, 2006.

[4] T. Hastie and C. Loader. Local regression: Automatic kernel carpentry. *Statistical Science*, 8(2):120–129, 1993.

[5] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–39, 2003.

[6] B. Matei and P. Meer. Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Trans. PAMI*, 28(10):1537–1552, 2006.

[7] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *Journal of Computational Geometry and Applications*, 14(4), 2004.

[8] S.Fleishman, D. Cohen-Or, and C. T.Silva. Robust moving least-squares fitting with sharp features. In *Proc. ACM SIGGRAPH*, 2005.

[9] G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *Intl. Conf. on Computer Vision*, 1993.

[10] R. Unnikrishnan, J.-F. Lalonde, N. Vandapel, and M. Hebert. Scale selection for the analysis of point-sampled curves. In *Proc. 3DPVT*, 2006.

[11] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[12] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, 1994.

[13] J. Wang, M. M. Oliveira, and A. E. Kaufman. Reconstructing manifold and non-manifold surfaces from point clouds. In *Proc. IEEE Visualization*, 2005.

[14] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004.