

2008

Alternative Parametric Boundary Reconstruction Method for Biomedical Imaging

Joseph Kolibal
University of Southern Mississippi

Daniel Howard
QinetiQ PLC

Follow this and additional works at: https://aquila.usm.edu/fac_pubs



Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Kolibal, J., Howard, D. (2008). Alternative Parametric Boundary Reconstruction Method for Biomedical Imaging. *Journal of Biomedicine and Biotechnology*, 2008.
Available at: https://aquila.usm.edu/fac_pubs/1389

This Article is brought to you for free and open access by The Aquila Digital Community. It has been accepted for inclusion in Faculty Publications by an authorized administrator of The Aquila Digital Community. For more information, please contact Joshua.Cromwell@usm.edu.

Research Article

Alternative Parametric Boundary Reconstruction Method for Biomedical Imaging

Joseph Kolibal¹ and Daniel Howard²

¹ Department of Mathematics, College of Science and Technology, The University of Southern Mississippi, Hattiesburg, MS 39406-0001, USA

² QinetiQ PLC, Malvern, Worcestershire WR14 3PS, UK

Correspondence should be addressed to Joseph Kolibal, joseph.kolibal@usm.edu

Received 2 September 2007; Accepted 15 January 2008

Recommended by Halima Bensmail

Determining the outline or boundary contour of a two-dimensional object, or the surface of a three-dimensional object poses difficulties particularly when there is substantial measurement noise or uncertainty. By adapting the mathematical approach of stochastic function recovery to this task, it is possible to obtain usable estimates for these boundaries, even in the presence of large amounts of noise. The technique is applied to parametric boundary data and has potential applications in biomedical imaging. It should be considered as one of several techniques to improve the visualization of images.

Copyright © 2008 J. Kolibal and D. Howard. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Three-dimensional (3D) computer reconstruction of a target volume or of a surface is an important activity in modern biomedical imaging. The accurate anatomical reconstruction in trauma or for use in image-guided intervention relies on mathematical imaging technology; and this paper develops the mathematical technique of stochastic function recovery [1] and illustrates its use for noisy boundary reconstruction. This is an alternative approach to the standard polynomial-based methods that we see as an add-on or complement to other techniques in use or being developed to improve upon the reconstruction of noisy boundary data to provide enhanced biomedical visualization.

The ability to distinguish features related to boundaries is intrinsic to technology of visualization. For example, in MRI imaging, a range of specialized methods have been developed for extracting information from signals so as to reconstruct images representing internal body structures [2]. Boundary recovery techniques apply to complex surgical procedures as with electroanatomical mapping that tracks the position of catheters inside the body with sparse signals recorded from electrodes at the tip of the catheter. Resulting surface maps must integrate real-time measurements with

preoperative MR or CT images, and account for mapping data errors in registration and error due to patient movement [3]. In general, when signals are affected by noise, it must be effectively removed in order to improve the visualization and compared with other medical imaging modalities, ultrasound images suffer from speckle noise that often makes for weak or incomplete boundaries [4].

Our approach is to use stochastic convolution-deconvolution operators [1, 5–7], which have useful statistical properties, to smooth noisy surface data in a manner that does not obliterate detail, and which effectively removes Gaussian noise. The motivation for the approach is that, intrinsically, stochastic interpolation using probabilistic kernels for the generating function of the row space of the linear operator performs well at removing noise when used to approximate data. However, the difficulty in applying these methods directly is that they bias the data to a mean of zero.

In approximating one-dimensional data, this is not usually a difficulty. However, in approximating multidimensional data, this can cause an apparent shift in the approximant when working in parametric coordinates. This is because the Gaussian kernels used smooth positive values to their mean value, thus potentially shifting the coordinates nearer the origin. When we approximate in one dimension,

the data values are shifted to the mean, however, the coordinates are not touched. If we are using parametric data, the coordinates are the data values ; and this means that the data can be shifted in (x, y) or (x, y, z) space. The less data existing or the greater the smoothing, the more will be this shift.

For example, when data is taken from a circle centered on the origin, the approximating curve is a circular curve centered on the origin, but of smaller radius, and the greater the smoothing applied in the approximation (if the data is very noisy), the smaller the area circumscribed by the approximating curve. As the number of sample points is increased, the approximation improves. However, for coarse surface data that is only smoothly varying, this can cause difficulties.

One approach is to make use of stochastic interpolation. Creating a dense noisy data set from the sparse noisy data using interpolation is followed by approximating this fine data set to recover the smoothed surface curve, thereby mitigating the shifting of the mean. While workable, this approach has several disadvantages, most notably that it requires a more costly interpolation step. It also requires the application of the technique more than once, and in the second or subsequent applications, the approximation must be done on a fine data set, meaning that many more points require approximation, again incurring a larger computational cost.

The solution is to make use of the convolution-deconvolution properties of stochastic interpolation combining the densification step with the approximation step. This would still be expensive. However, we introduce an approximate means for doing the interpolation that interpolates for smooth data, but which approximates for noisy data, thus avoiding the costly need to construct the inverse operator needed for interpolation.

2. DEVELOPMENT

Consider the task of sampling a known function $f(u)$ at points $f(u_k) = v_k$ with $u_k \in [0, 1]$ so as to determine its value at $x_j \in [0, 1]$. The *stochastic interpolant* [5] to the data $\{(u_k, v_k)\}$, $k = 1, \dots, n$ is given by

$$B_{mn}A_{nn}^{-1}v, \quad (1)$$

where $v = (v_1, v_2, \dots, v_n)^T$ is the data vector, and where A_{nn} is a row stochastic matrix whose coefficients consist of the $n \times n$ values a_{jk} . Choosing the generator of the row space of A_{nn} to be the Bernstein functions [1] (named after Bernstein as the derivation of this form that can be obtained from the Bernstein polynomials), we have

$$a_{jk} = \frac{1}{2} \left[\operatorname{erf} \left(\frac{u_{k+1} - x_j}{\sqrt{\sigma n}} \right) + \operatorname{erf} \left(\frac{x_j - u_k}{\sqrt{\sigma n}} \right) \right] \quad (2)$$

with $\sigma > 0$ on the partition $w_k = (u_k + u_{k+1})/2$, with $w_0 = -\infty$ and $w_n = \infty$ yielding a stochastic matrix. Setting $x_j = u_j$ generates the entries of A_{nn} , and setting x_j to any set of m consecutive values in $[0, 1]$ generates the coefficients of B_{mn} , that is, the coefficients of B_{mn} are constructed in the same

manner as for A_{nn} , except that the nodes x_j at which b_{jk} is evaluated may differ from the values at which the data are given. While any probability density function (pdf) can be used, appropriately replacing the mean and variance of the Gaussian in (2), a pdf based on the normal distribution is consistent with the problem of filtering Gaussian noise.

In stochastic interpolation, with the coefficient of A_{nn} generated by (2), we can interpret A_{nn}^{-1} as the discrete deconvolution of the data yielding the preimage generated by $A_{nn}^{-1}v$. This preimage is then convolved by B_{mn} to yield an m -vector of values that interpolates the data v at the output coordinates x_j , $j = 1, \dots, m$; the output coordinates are those that were used to generate the coefficients of B_{mn} . It is for this reason that we have elected to represent the matrix A using another symbol B since it is desirable to emphasize its role in convolution.

Defining $w_k = (u_k + u_{k+1})/2$ with $w_0 = -u_0/2$ and $w_n = 1 + u_n/2$ with σ constant yields a coefficient structure in which A_{nn} is a diagonal matrix times a symmetric Toeplitz matrix: $A_{nn} = DT_{nn}$. Inversion or solution of these matrices can be accomplished in $\mathcal{O}(n^2)$ operations [8], however in the cases of interest to us, this is not necessary. It is possible to do better than this using an approximate inverse in which the row space of $A_{nn}^I \approx A_{nn}^{-1}$ is generated directly. While evaluating $B_{mn}A_{nn}^I$ is still $\mathcal{O}(n^2)$, it is significantly faster than the Toeplitz matrix inversion of A_{nn} to obtain A_{nn}^{-1} .

The approximate inverse is an approximation precisely because $A_{nn}A_{nn}^I \neq I_{nn}$, that is, in applying stochastic interpolation to the data v , there is an error:

$$e_0 = v - A_{nn}A_{nn}^I v = (I_{nn} - A_{nn}A_{nn}^I)v. \quad (3)$$

Thus the interpolant to the data can be expressed using successive correction to the errors using

$$\begin{aligned} v &= A_{nn}A_{nn}^I v + e_0 \\ &= A_{nn}A_{nn}^I v + AA_{nn}^I e_0 + e_1 \\ &\vdots \\ &= A_{nn}A_{nn}^I v + A_{nn}A_{nn}^I e_0 + A_{nn}A_{nn}^I e_1 \\ &\quad + \dots + A_{nn}A_{nn}^I e_p + e_{p+1}. \end{aligned} \quad (4)$$

Substituting for e_k , from $k = 0$ to p , gives

$$v = A_{nn}A_{nn}^I \left(\sum_{k=0}^p (I_{nn} - A_{nn}A_{nn}^I)^k \right) v + e_{p+1}, \quad (5)$$

and truncating the sum gives a working formula in which $\tilde{v} \approx v$ even for larger values of σ , so that the method nearly interpolates. Truncating at $p = 2$ and applying the formula to generate m output values instead of n gives

$$\tilde{v} = B_{mn}A_{nn}^I (I_{nn} + G_{nn} + G_{nn}^2)v, \quad (6)$$

where $G_{nn} = (I_{nn} - A_{nn}A_{nn}^I)$. Provided that σ in (2) is small, the error in constructing A_{nn}^{-1} using A_{nn}^I is small, however if larger values of σ are used, then greater smoothing is applied to the data and the use of (6) becomes necessary when σ is

larger than 0.05. However, it will become apparent that it is because of this greater smoothing that it is unnecessary to apply any corrections as shown in (6), and thus the direct computation of $B_{mn}A_{nn}^I v$ is found to be convenient and efficient, requiring only a single matrix multiply.

In working with stochastic data recovery, it is obvious that $B_{mn}v$ using the Bernstein functions mollifies the data vector v , and thus provides an approximation vector of length m to the initial vector v of length n . Consider the evaluation of $B_{mn}A_{nn}^{-1}v$ where the generator of the row space of A_{nn} and B_{mn} are given by (2). This interpolates the data provided that σ_B , meaning that the variance of the Gaussian pdf that is used to generate B_{mn} , is the same as the variance σ_A that is used to generate A_{nn} . If instead $\sigma_B > \sigma_A$, then the preimage $A_{nn}^{-1}v$ will be oversmoothed when B_{mn} is applied to the preimage, and the result will be approximation. Similarly, if $\sigma_B < \sigma_A$, then the preimage $A_{nn}^{-1}v$ will not be smoothed sufficiently, and the data will be roughened, or more appropriately it will be deconvolved.

With statistical errors present in multidimensional data, interpolation in parametric coordinates may yield an extremely complex curve, and the errors may cause the curve to wiggle excessively, often crossing over on itself. Clearly, some form of smoothing is necessary. However, as noted in Section 1, simply approximating the data so as to smooth these errors may introduce translation errors in the approximating surface representation, particularly when the number of data points is small, or the smoothing specified by σ is large. Since the approximation is convergent, a simple work-around can be achieved by densification of the data by interpolation as this will minimize this shift on subsequent smoothing. This leads to a computationally efficient approach to surface recovery that avoids translation errors while smoothing the noise based on evaluating $B_{mn}A_{nn}^I v$, where m is the desired number of output points representing the boundary, and n are the number of input data points and $\sigma_B > \sigma_A$.

To demonstrate this form, note that the intended construction is to evaluate $B_{pn}A_{nn}^{-1}v$, where p is significantly greater than n with $\sigma_B = \sigma_A$, and then applying smoothing to this densified data by multiplying by C_{mp} , where $\sigma_C > \sigma_A$. In effect, this interpolates the data and then smooths it by the application of the approximation C_{mp} to the densified data. Thus a two-step algorithm can be described as follows:

- (1) compute densified interpolant $B_{pn}A_{nn}^{-1}v$ to the data vector v ;
- (2) compute boundary approximant $C_{mp}(B_{pn}A_{nn}^{-1}v)$.

This algorithm is equivalent to the following:

- (1) compute densified boundary approximate interpolant $B_{mn}A_{nn}^I v$, to the data vector v ;

that is, there exist σ_C and σ_B such that applying C_{mp} to $B_{pn}A_{nn}^{-1}v$ is the same, or nearly the same, as applying B_{mn} to $A_{nn}^I v$. The use of A_{nn}^I for a wide range of σ_A instead of A_{nn}^{-1} introduces some additional smoothing, allowing for less smoothing to be used on the convolution step, that is, when applying B_{mn} . Since it saves an unnecessary matrix multiply, it is clearly faster. The construction of A_{nn}^I is not difficult,

remarkably being given by the inverse of the generator of the row space of A_{nn} . The elements of the approximate inverse are given by the reciprocals of the coefficients of A_{nn} . It is for this reason that the direct inversion of A_{nn} , or solution of the system using efficient Toeplitz solvers, is not needed. The only exception may be when the data is free of noise and exact interpolation without any smoothing is desired.

The reconstruction of surface data is done using a parametric representation $\{s_i\}_{i=1}^n$ in which $s_i = (x(t_i), y(t_i))$ for two-dimensional data, and $s_i = (x(t_i), y(t_i), z(t_i))$ for three-dimensional data, where $0 \leq t_i \leq 1$. The recovery of the data is done using (1) or its modification using an approximate inverse A_{nn}^I , applied successively to the data pairs $\{(t_i, x_i)\}$, $\{(t_i, y_i)\}$ in two dimensions and to the data triplets $\{(t_i, x_i)\}$, $\{(t_i, y_i)\}$, $\{(t_i, z_i)\}$ in three dimensions. For example, in the case of interpolating two-dimensional data, we apply $B_{mn}A_{nn}^{-1}x$ and $B_{mn}A_{nn}^{-1}y$ to obtain the interpolant to the position vector x and the position vector y , or we apply $B_{mn}A_{nn}^I x$ and $B_{mn}A_{nn}^I y$ to obtain the approximate interpolants.

In reconstructing a parametrically represented surface generated from image data from pixel values, for instance, an algorithm for boundary detection and sorting is necessary. In our analysis, it is assumed that this is available, however the errors generated in parameterizations of the surface may not be entirely random, and thus systematic errors in surface representation will also be introduced. For the purpose of assessing the performance of the boundary recovery algorithms, it will be assumed that the errors are random Gaussian with a mean of zero, with variable variances v , generated using the random variable

$$\xi = v\sqrt{-2 \log(r_1)} \cos(2\pi r_2), \quad (7)$$

where r_1 and r_2 are two random numbers uniformly distributed in the interval $[0, 1]$. Thus the parametrically ordered data $\{(x_1(t_i), \dots, x_d(t_i))\}$, with $d = 2$ or 3 , and $i = 1, \dots, n$, are perturbed to yield the data sets $\{(x_1(t_i) + (\xi_1)_i), \dots, x_d(t_i) + (\xi_d)_i)\}$.

In applying stochastic data recovery to the problem of finding the shape of a parametrically defined boundary, the problem of closed curves needs to be addressed. In the presence of large amount of noise, the two endpoints of the parametrically defined curves may not match: while ideally $(x_1(t_1), \dots, x_d(t_1)) = (x_1(t_n), \dots, x_d(t_n))$ in the case of a close loop, in the presence of errors this will not be the case.

Finally, in implementing the algorithm, it was found that the dependence of (2) on \sqrt{n} in the denominator made the choice of σ dependent on n , as the boundary data density increased, the recovery using any given value of σ produced increasingly rougher curves as n increased, and thus it was found convenient to evaluate a_{jk} and b_{jk} based on a constant value of σ . Additionally, the algorithm was applied to all of the boundary data associated with a parametric data set to construct the boundary curve, rather than decomposing the data into overlapping blocks. The merits of using all of the data are a slight improvement in accuracy, while the drawbacks are that the cost of evaluating the algorithm increases as the block size increases, and this should be born

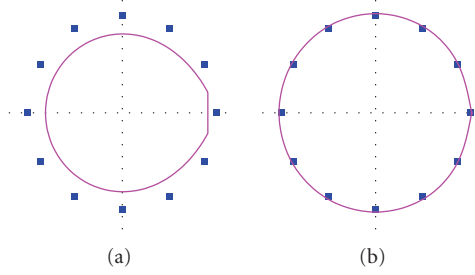


FIGURE 1: Recovery of the boundary of disk specified by 12 points, starting at $(1,0)$ on the x -axis, and moving counterclockwise around the circle. Note that in the circle in (a) the boundary of the recovered disk is approximated with $\sigma_B = 0.001$ using $B_{100,12}\nu$, resulting in the inscribed disk, while in (b) the boundary is constructed using approximate interpolation using $\sigma_A = \sigma_B = 0.001$ and applying $B_{100,12}A_{12,12}^T\nu$.

in mind in applying the algorithm to large complex three-dimensional data sets.

The choice of σ depends on the smoothness of the desired boundary curve. The larger is the value of σ , the smoother are the results. The values for σ_A and σ_B depend on the amount of noise, as well as on the presumed smoothness of the boundary data. While this seems to present difficult choices, it is less complicated than it appears, as the choice for σ_A will usually be any sufficiently small value. For example, setting $\sigma_A \leq 10^{-6}$ allows for representing the data as nearly piecewise linear along the boundaries. In contrast, the choice for σ_B requires some evaluation as this determines the smoothness of the recovered boundary.

3. RESULTS AND DISCUSSION

We begin by applying the process to the recovery of the boundary of a disk defined parametrically by 12 uniformly distributed points with no random errors in the data as shown in Figure 1. The figure clearly illustrates the aforementioned difficulty of attempting to approximate (smooth parametric data).

In examining Figure 1(a), it is also important to observe that the beginning and the end of the curve are joined by a straight line in this example. The reason is that the slope of the approximant to the data $(x(t_1), y(t_1)), (x(t_2), y(t_2)), \dots$ is not the same as the slope to the data $\dots, (x(t_{n-1}), y(t_{n-1})), (x(t_n), y(t_n))$, and thus in this example, the first approximated point (r_1, s_1) does not agree with the last approximated point (r_m, s_m) , and are joined graphically with a straight line to close the curve.

A solution to the mismatch is to overlap the curves during reconstruction, that is, at several points away from the last point, and ending the construction several points away from the first point, then only using the curve from the first to the last point. In all of the studies presented, there is no attempt to overlap the curves in order to magnify these boundary affects, and to demonstrate that they are mostly negligible, as seen in Figure 1(b), whenever the construction is done correctly. For large data sets where it may be

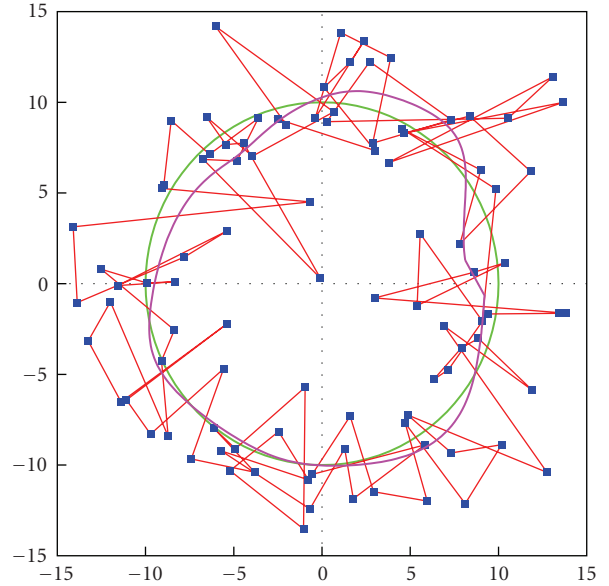


FIGURE 2: Recovery of the boundary of disk specified by 96 points with Gaussian noise of $\nu = 3$, starting at $(1,0)$ on the x -axis and moving counterclockwise around the circle with the boundary recovered using approximate interpolation, with $\sigma_A = 1 \times 10^{-6}$ and $\sigma_B = 1 \times 10^{-4}$. The recovered curve is shown as a thicker line.

computationally advantageous to block the data, overlapping the endpoints is readily accomplished.

It is important to realize that if the number of points representing the disk were to be much more than 12, then it would have been difficult to visualize the contraction of the recovered boundary curve since the approximation is convergent. Thus for a sufficiently large number of data points, the difference between the approximating curve and the curve itself becomes arbitrarily small.

Recovering the boundary of a disk becomes more difficult, as shown in Figure 2, particularly if the amount of noise is quite large. In this example, the Gaussian noise for a disk of radius 5 is specified as $\nu = 3$ yielding an extensively scattered data set. While the level of noise in this illustrative example is much higher than would be expected in any realistic imaging situation, the example serves a two-fold purpose: (1) on the one hand it shows the robustness of the method at recovering a reasonable representation of the surface from the data that is more consistent with noise than data, and (2) it shows that the effects of errors in constructing the parameterization are less of an issue than might be presumed. In the figure, the connectivity of the boundary data is not ordered in θ moving counterclockwise around the circle, and so it is quite likely that any minor errors in parameterization, for example, using even the simplest unconstrained nearest neighbor search of the data, would cause unrecoverable errors in the surface representation that is recovered.

As expected, reducing the noise to $\nu = 1/2$ significantly improves the recovery, even for half as many points, as shown in Figure 3.

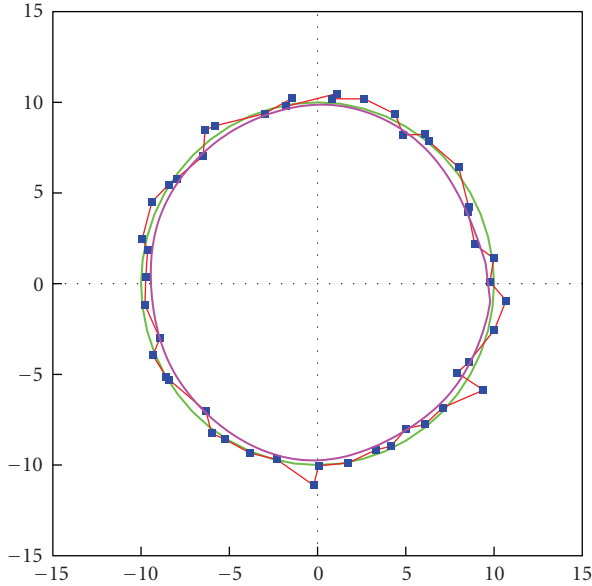


FIGURE 3: Recovery of the boundary of disk specified by 48 points with Gaussian noise of $\nu = 1/2$ as in Figure 2.

While both of these figures are typical, the symmetry and the smoothness of the boundary of the disk make the recovery somewhat less challenging than for a more complex geometrical shape. Thus the performance of the algorithm is examined on a star-shaped region generated from the vertex data $\{(8,65), (72,65), (92,11), (112,65), (174,65), (122,100), (142,155), (92,121), (42,155), (60,100), (8,65)\}$. Note that the recovery makes use of σ that on both steps is the same as that used in recovering the boundary of the disk.

The shape of the star in Figure 5 has become more wiggly using the smoothing parameters the same as in the case of lesser noise. The problem is a classical one: there is no means to discern the shape of the figure from the noisy data, except to note that an acceptable shape is determined by the smoothness of the boundary that is intrinsic to the figure. In this case, changing the smoothing can accommodate this subjective assessment, as illustrated in Figure 6. Note that the recovered boundary is consistent with the curve recovered from the less noisy data set: compare Figures 4 and 6 as shown in Figure 7.

The effects of doubling the smoothing by taking σ_B to be twice as large are clearly evident. Since the noise in both cases was generated using the same seed, it is only the magnitude of the excursions away from the star's boundary that change, and hence the figures are directly comparable. This perhaps most plainly illustrates that interaction between noise and smoothing, as the two curves are nearly identical. Even when the noise is doubled again to $\nu = 16$, the shape of the recovered curve using $\sigma = 0.0004$ is remarkably consistent, as shown in Figure 8. At this level of noise there is some loss of resolution of the limbs, however the recovered boundary is recognizable as being related to the two boundaries obtained in Figures 4 and 6. While this is artificial in that the amount of noise in the data in real problems is not known, it does

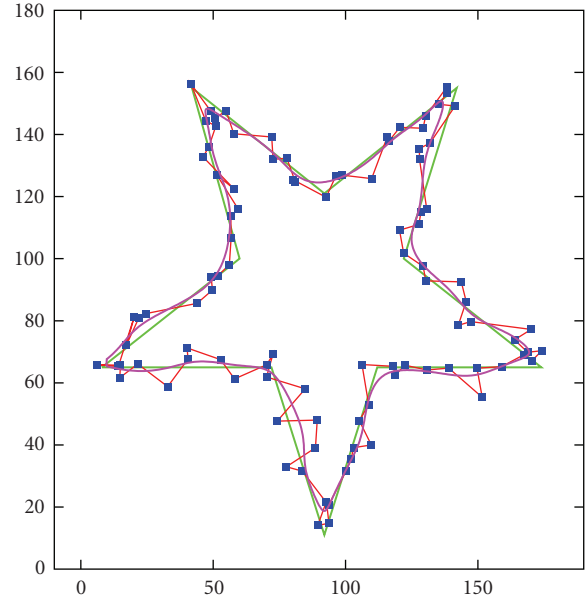


FIGURE 4: Recovery of the boundary of the star specified by 100 points: 10 along each arm with Gaussian noise of $\nu = 4$. The boundary is constructed using approximate interpolation, using $\sigma_A = 1 \times 10^{-6}$ and $\sigma_B = 1 \times 10^{-4}$. The recovered curve is shown as a thicker line.

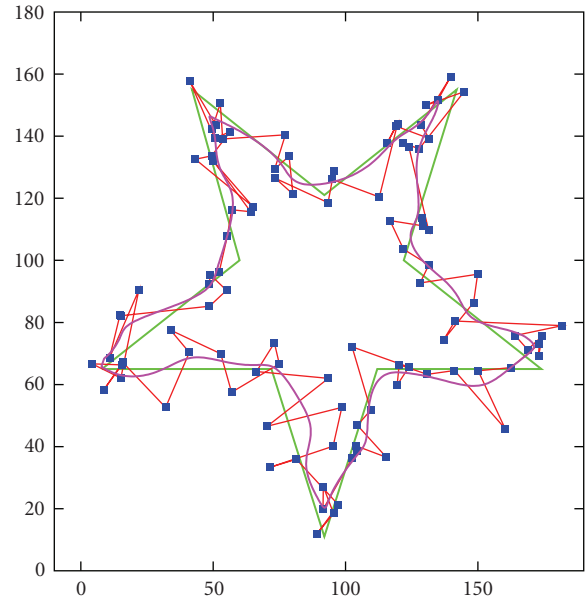


FIGURE 5: Recovery of the boundary of the star specified by 100 points with Gaussian noise of $\nu = 8$ as in Figure 4. The recovered curve is shown as a thicker line.

demonstrate the robustness of the algorithm at consistently recovering the boundary data irrespective of the added noise.

A final remark on the computing of the approximate interpolant is the following. Since the approximate interpolant fails to interpolate when σ is large or, more appropriately, fails to interpolate rapidly varying data, quite some effort was expended in developing mechanisms for

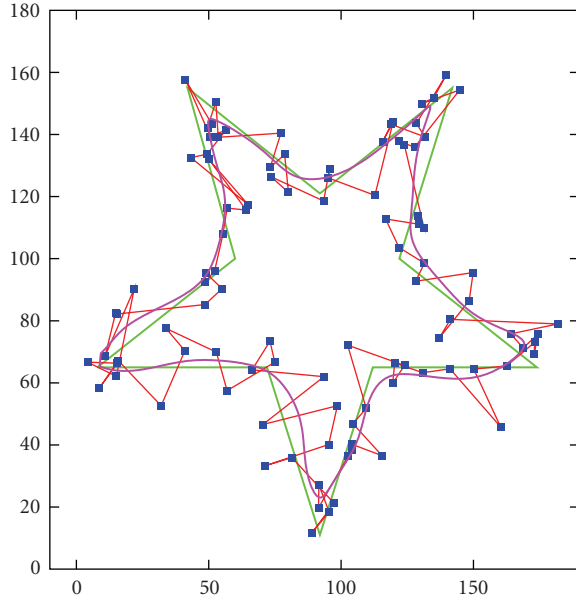


FIGURE 6: Recovery of the boundary of the star specified by 100 points with Gaussian noise of $\nu = 8$ as in Figure 5. The boundary is constructed using approximate interpolation with $\sigma_A = 1 \times 10^{-6}$ and $\sigma_B = 2 \times 10^{-4}$. The recovered curve is shown as a thicker line.

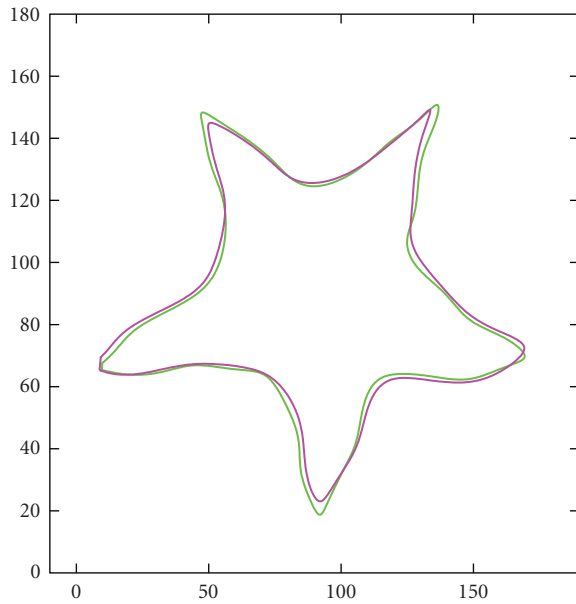


FIGURE 7: Recovery of the boundary of the star comparing the recovered boundary shown in Figure 4 to the recovered boundary shown in Figure 6. In this comparison, the recovery algorithm is implemented so that the smoothing is proportional to the noise in the cases being compared, yielding excellent results.

being able to use large σ for smoothing and yet still maintain some fidelity with the boundary data while constructing the preimage. As it developed, this iterative correction was not necessary: the algorithm performed well even without the introduction of any corrections. In part, this is due to the

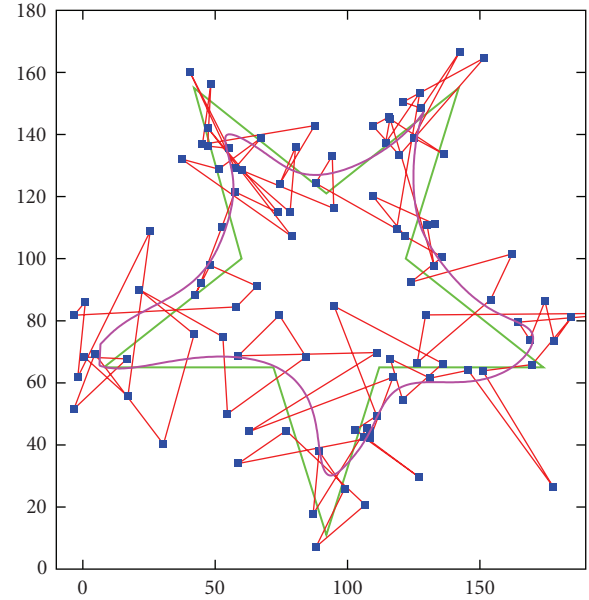


FIGURE 8: Recovery of the boundary of the star specified by 100 points with Gaussian noise of $\nu = 16$. In this case, an extremely large amount of noise is added, however the algorithm is effective at recovering features of the star shape. The boundary is constructed using approximate interpolation by using $\sigma_A = 1 \times 10^{-6}$ and $\sigma_B = 4 \times 10^{-4}$. The recovered surface is shown as a thicker line.

rather simple shapes, and the relatively large amounts of noise that were examined.

In the case when a surface is oscillating rapidly with the noise much less than this surface oscillation, it is clearly necessary to implement the algorithm using this additional correction. Indeed, given sufficiently rough surface data, it may be necessary to use stochastic interpolation as otherwise some high-frequency surface details will be oversmoothed by the approximate inverse construction.

4. CONCLUSIONS

Examination of several attempts at domain boundary reconstruction in two dimensions is encouraging using the stochastic data recovery techniques. In particular, qualitative assessments demonstrate the viability of utilizing stochastic function recovery methods for reconstructing parametrically defined edges. Since the approach is intrinsically one dimensional, its extension to three dimensions is not difficult, and thus can easily be implemented and tested on more realistic problems. Moving to three dimensions poses no additional cost other than that more data has to be processed, that is, the algorithmic costs scale directly with the number of lines being evaluated, and the number of points on each line at which the algorithm is applied.

It should be noted that the proposed technique does not solve all problems involving noisy surface reconstruction, however it does provide an additional tool for analysis. The Gaussian-based kernels (the Bernstein functions) which were used to generate the row space of the matrices, were remarkably effective at cancelling the noise even under

the most extreme conditions where the noise essentially obliterated the image shape. Of course, this noise was Gaussian and so it is only reasonable that the proposed approach would work well in these circumstances. For other types of noise, the use of alternative probability density functions for generating the mollifiers is easily accomplished, and thus the method has substantial design flexibility, and these options need to be explored in detail to ascertain their utility at cancelling these other types of noise.

The computational advantages of the technique are that it requires only two matrix multiplies of the data vector, and thus the approach is relatively cost effective. Moreover, the method is easily implemented in parallel, and further computational gains in efficiency can be achieved by blocking the data since typically only a segment of the entire data vector is needed to recover the data in any region. If fixed block sizes can be implemented, then the cost of the two matrix vector multiplies can be further reduced as the matrix-matrix multiply needs to be done only once, and so the algorithm reduces to a single-block matrix-vector multiply.

REFERENCES

- [1] J. Kolibal and D. Howard, "The novel stochastic Bernstein method of functional approximation," in *Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems (AHS '06)*, pp. 97–100, Istanbul, Turkey, June 2006.
- [2] I. Kastanis, S. R. Arridge, A. M. S. Silver, D. L. Hill, and R. Ravazi, "Reconstruction of the heart boundary from under-sampled cardiac MRI using Fourier shape descriptors and local basis functions," in *Proceedings of the 2nd IEEE International Symposium on Biomedical Imaging: Macro to Nano (ISBI '04)*, vol. 2, pp. 1063–1066, Arlington, Va, USA, April 2004.
- [3] Z. Malchano, "Image guidance in cardiac electrophysiology," Massachusetts Institute of Technology, Cambridge, Mass, USA, 2006.
- [4] J. Xie, Y. Jiang, and H.-T. Tsui, "Segmentation of kidney from ultrasound images based on texture and shape priors," *IEEE Transactions on Medical Imaging*, vol. 24, no. 1, pp. 45–57, 2005.
- [5] D. Howard and J. Kolibal, "Image analysis by means of the stochastic matrix method of function recovery," in *ECSIS Symposium on Bio-Inspired, Learning, and Intelligent Systems for Security (BLISS '07)*, pp. 97–101, Edinburgh, UK, August 2007.
- [6] J. Kolibal and D. Howard, "Implications of a novel family of stochastic methods for function recovery," in *Proceedings of the 2nd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '06)*, pp. 495–498, Pasadena, Calif, USA, December 2006.
- [7] J. Kolibal and D. Howard, "MALDI-TOF baseline drift removal using stochastic bernstein approximation," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 63582, 9 pages, 2006.
- [8] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices," *SIAM Journal on Applied Mathematics*, vol. 12, no. 3, pp. 515–522, 1964.