A Performance Comparison of Piecewise Linear Estimation Methods

Manjula A. Iyer, Morgan M. Harris, and Layne T. Watson
Departments of Computer Science and Mathematics
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061
Email contact: manjula@vt.edu

Michael W. Berry
Department of Computer Science
University of Tennessee
Knoxville, Tennessee 37996

Abstract—**The response functions in many engineering problems are piecewise smooth functions, although globally they can be highly nonlinear. The linear Shepard algorithm (a moving window weighted least squares method based on linear functions) usually creates reasonable approximations. When used to approximate data obtained from piecewise linear functions, a better approximation near the function creases is obtained when a robust linear Shepard algorithm is used. Due to limitations of the robust linear Shepard algorithm in high dimensions, RIPPLE (residual initiated polynomial-time piecewise linear estimation) was developed for piecewise linear estimation of high dimensional data. RIPPLE selects minimal sets of data based on a minimal residual criterion. Once the best minimal set is obtained, RIPPLE adds other consistent data points using a robust statistical procedure. RIPPLE is resistant to outliers. It produces good approximations even if the data contains significant random errors. The code L2WPMA (least squares weighted piecewise monotonic approximation, ACM TOMS Algorithm 863) calculates a piecewise monotonic approximation to n univariate data points contaminated by random errors. The continuous piecewise linear approximation consists of k (a positive integer provided by the user) monotonic linear splines, alternately monotonically increasing and monotonically decreasing. An optimal approximation is obtained if there are at most k monotonic sections in the exact data. This paper will present a theoretical complexity comparison and an empirical performance comparison of the linear Shepard algorithm, robust linear Shepard algorithm, RIPPLE, and L2WPMA for univariate datasets.**

1. INTRODUCTION

Piecewise linear approximation methods for univariate datasets are used in a wide range of applications including statistics, econometrics, management science, and psychology. Furthermore, many engineering models, while highly nonlinear globally, are approximately piecewise linear, making it possible to construct high quality locally linear interpolants. Approximation near function creases (also known as turning points or peaks) has attracted much research interest in spectroscopy, chromatography, signal processing, and image processing.

Shepard's interpolation method [13], based on a weighted average of values at data points, usually creates reasonable approximations. The linear Shepard method [6] is a moving window least squares method based on linear functions. It is best used for approximation in high dimensional sparse data sets as it requires fewer data points than other Shepard variations (such as quadratic, cubic, trigonometric, etc.) to construct the fit. When used to approximate data obtained from piecewise linear functions, its performance degrades near the function creases. The performance can be improved by using M-estimation with the linear Shepard method (called the robust linear Shepard method) [8]. This method has limitations in high dimensions and is also not specifically designed to approximate data obtained from piecewise linear functions. The robust linear Shepard method can be used instead of the standard linear Shepard method when the data is known to contain significant errors. RIPPLE [9] (residual initiated polynomial-time piecewise linear estimation) is a piecewise linear approximation method. It produces robust estimates in polynomial time and performs well in high dimensions even if the data contains significant random errors. L2WPMA (least squares weighted piecewise monotonic approximation, ACM TOMS Algorithm 863) [2] calculates a piecewise monotonic approximation to n univariate data points contaminated by random errors. The continuous piecewise linear approximation consists of k (a positive integer provided by the user) monotonic linear splines, alternately monotonically increasing and monotonically decreasing. L2WPMA constructs an optimal approximation if there are at most k monotonic sections in the exact (uncontaminated) data.

This paper presents a performance comparison of the linear Shepard algorithm, robust linear Shepard algorithm, RIPPLE, and L2WPMA for univariate data. The remaining sections of the paper are organized as follows. The next section describes the linear Shepard and robust linear Shepard methods. Section 3 describes RIPPLE. L2WPMA (TOMS Algorithm 863) is summarized in Section 4. Section 5 presents sample test function results, which are followed by concluding remarks in Section 6.

2. LINEAR SHEPARD ALGORITHM

Let E^m denote m-dimensional Euclidean space, $x = (x_1,\ldots,x_m) \in E^m$, and for real w let $w_+ = \max\{0, w\}$. The scattered data interpolation problem can be defined as: given a set of irregularly distributed points $x^{(i)} \in E^m$, $i = 1,\ldots, n$, and scalar values f_i associated with each point satisfying $f_i = f\big(x^{(i)}\big)$ for some underlying function $f : E^m \to E$, look for an interpolating function $\tilde{f} \approx f$ such that $\tilde{f}\big(x^{(i)}\big) = f_i$. The Franke and Neilson [5] modification of the original Shepard algorithm [13] defines an approximation to $f(x)$ by

$$\tilde{f}(x) = \frac{\sum_{k=1}^{n} W_k(x) P_k(x)}{\sum_{i=1}^{n} W_i(x)},$$

where $P_k(x)$ is a local approximant to the function $f(x)$ centered at $x^{(k)}$, with the property that $P_k\big(x^{(k)}\big) = f_k$. $W_k(x)$ is the weight function and is of the form

$$W_k(x) = \left[\frac{\Big(R_w^{(k)} - d_k(x)\Big)_+}{R_w^{(k)} d_k(x)}\right]^2,$$

where $d_k(x) = \big\|x - x^{(k)}\big\|_2$ is the Euclidean distance between the points x and $x^{(k)}$, and the constant $R_w^{(k)} > 0$ is a radius of influence about the point $x^{(k)}$ chosen just large enough to include N_w points. The data at $x^{(k)}$ only influences $\tilde{f}(x)$ values within this radius.

In polynomial variations of Shepard's algorithm, the function P_k is written as a Taylor series about the point $x^{(k)}$ with constant term $f_k = P_k\big(x^{(k)}\big)$ and coefficients chosen to minimize the weighted square error

$$\sum_{\substack{i=1 \\ i\neq k}}^{n} \omega_i\big(x^{(k)}\big) \Big[P_k\big(x^{(i)}\big) - f_i\Big]^2,$$

with weights

$$\omega_i\big(x^{(k)}\big) = \left[\frac{\Big(R_p^{(k)} - d_i\big(x^{(k)}\big)\Big)_+}{R_p^{(k)} d_i\big(x^{(k)}\big)}\right]^2,$$

and $R_p^{(k)} > 0$ defining a radius about $x^{(k)}$ within which data is used for the least squares fit. R_w and R_p are computed as

$$R_w = \frac{D}{2}\sqrt{\frac{N_w}{n}}, \quad R_p = \frac{D}{2}\sqrt{\frac{N_p}{n}},$$

where $D = \max_{i,j} \big\|x^{(i)} - x^{(j)}\big\|_2$ is the maximum distance between any two data points, and N_w and N_p are arbitrary constants. Later versions of the algorithm defined $R_w^{(k)}$ ($R_p^{(k)}$) as the radius of the smallest ball centered at $x^{(k)}$ containing N_w (N_p) points.

The linear Shepard method is a moving window weighted least squares method based on linear functions. It requires fewer data points than other (higher order) Shepard algorithm variations to construct the fit and hence is more practical for use on sparse data in higher dimensions. However, for high dimensional piecewise linear functions, the performance of the linear Shepard algorithm degrades near function surface creases. This is so because the method uses weighted least squares to construct the fit. It uses all the nearby data points and "averages" all the facets near a ridge. This motivates the use of a robust linear Shepard algorithm near a function ridge. Iyer and Watson [8] proposed a robust linear Shepard algorithm using M-estimation. Using M-estimation with the linear Shepard method constructs better approximations than the standard linear Shepard method. However, the breakdown bound [7] for M-estimation is $1/(p+1)$, where p is the number of parameters to be estimated. The value of the breakdown bound for M-estimation is low, meaning that a large number of data points are required to obtain robust estimates. However, the primary advantage of the linear Shepard method is that in m-dimensional space, it requires $O(m)$ data points to construct the fit, which is no longer possible when M-estimation is used, since the number of points required by M-estimation is $O(m^2)$. (If F is the fraction of points that are outliers, $F(m+2) \leq \frac{1}{(m+2)}(m+2) \Rightarrow (m+2)^2 F \leq m+2$, so $O(m)$ outliers requires $O(m^2)$ data points.)

This computational complexity problem (the requirement for $O(m^2)$ data points) is solved by using a different robust estimation technique, least median of squares (LMS) [12], which has a breakdown value of 1/2, independent of the dimension m. Thus, LMS

would require $O(m)$ data points to construct a robust linear Shepard approximation. LMS achieves this remarkable statistical efficiency by constructing all $\binom{n}{p+1}$ possible fits to subsets of $p+1$ points, where again p is the number of parameters being estimated. For some n and p, this is practical, but in general this factorial complexity is computationally untenable. This motivates the use of a hybrid statistically robust method, somewhere between M- and LMS-estimation, that requires fewer data points than the M-estimation based linear Shepard method and yet produces better approximations than the standard linear Shepard method.

3. RIPPLE

This section explains the algorithm RIPPLE (residual initiated polynomial-time piecewise linear estimation)[9], which was designed to produce robust estimates of data obtained from high dimensional piecewise linear functions in polynomial time. The standard linear Shepard algorithm uses weighted linear least squares to construct the local approximation. It uses all the data to construct the local fit. As a result, it does not produce good approximations near a piecewise linear function ridge as it "averages" all the facets near the ridge. When M-estimation is used, the linear Shepard method ideally picks the facet that is consistent with a majority of the nearby data. Even though this is usually better than the standard linear Shepard method, there are cases when the required majority of data points may not lie on the same facet as the point of approximation. This produces large errors, which can be reduced if the points to be used in the fit are chosen carefully, as LMS estimation would have done.

Let $S = \{i_1, i_2, \ldots, i_{s_k}\}$ be the set of indices (where $s_k = N_p - 1$) corresponding to points that lie within $R_p^{(k)}$ of $x^{(k)}$, excluding $x^{(k)}$. The number of points N_p is chosen to be $2\lceil 3m/2 \rceil$. The most consistent points (not necessarily a majority) in the set S must be used for the local least squares approximation. Suppose that based on some criteria, the points in this set could be classified as 'inliers' and 'outliers' such that each inlier produces a lower absolute residual (vertical distance to the approximation) than all outliers. Intuition suggests that if a point is classified as an inlier, the points close to it have a higher probability of being classified as an inlier. However, if two points that have equal distance from an inlier are on two different facets, then the corresponding function values are "averaged", resulting in a large approximation error. Thus, it is not possible to pick the inliers based solely on the criterion of distance from the point of approximation x. Also, it is not possible to predict how many data points will be used for constructing the local approximation.

Suppose that the best set (set of points producing minimum sum of squared residuals) of the minimum number of points required to construct the fit is chosen. It is now possible to examine every other data point and determine whether it should be added to the best set, based on the residual that is produced. Thus, the problem is now reduced to finding the best minimal set of points. The idea for choosing minimal sets of points required to construct the fit has been borrowed from the RANSAC (random sample consensus) algorithm [4], which chooses the minimal sets of points at random. If the data dimension is m, the number of parameters in $P_k(x)$ to be estimated is m, requiring at least $(m+1)$ number of data points to construct a linear approximation. The total number of data points is s_k. The best minimal set can be chosen in $\binom{s_k}{m+1}$ ways. However, in general this has exponential complexity and is therefore untenable.

The best minimal set of points can be approximated in polynomial time by using a special distance criterion that is described as follows. Define the $s_k \times (m+3)$ distance matrix D such that $D_{.1} = (i_1, i_2, \ldots, i_{s_k})^t$ and $D_{i,j+1}$ is the index of the point that has least Euclidean distance from the point whose index is $D_{i,j}$, for $i = 1$ to s_k and $j = 1$ to $m+2$, subject to the constraint that every row of D must contain distinct indices. In case $x^{(i_r)}$ and $x^{(i_s)}$ are equidistant from $x^{(D_{i,j})}$, choose the one closer to $x^{(k)}$. If that choice also results in a tie, use index i_r (assuming $r < s$).

Compute the local approximation $P_k(x)$ using $x^{(k)}$ and different sets of $(m+1)$ distinct points $x^{(i)}$ that are picked as follows. For each row t, $t = 1$ to s_k, in D, pick the point corresponding to index $D_{t,1}$. The remaining m points can be picked in $\binom{m+2}{m}$ ways from row t. Thus, the total number of local approximations $P_k(x)$ computed using $x^{(k)}$ and a set of $(m+1)$ other points will be $s_k\binom{m+2}{m}$. For each approximation, compute the absolute residuals (vertical distances) for the $(m+1)$ points used to compute the least squares fit. Record which set of $(m+1)$ points produces the minimum least squares error. Let $R = \{i_1, i_2, \ldots, i_{m+1}\}$ correspond to the indices of these $m+1$ points. In case of a tie for this best set, choose the set containing the points closer to $x^{(k)}$ (precisely, sort the distances $\|x^{(i_j)} - x^{(k)}\|$ into an $(m+1)$ vector in increasing order, then compare the two vectors lexicographically; if these distances are equal, sort and compare lexicographically the sets of indices).

Once the best set of minimal points is obtained, compute the linear interpolant at $x^{(k)}$. Compute

the average residual produced by the points $x^{(i)}$, $i \in R$, and record the coefficients of the linear interpolant. Use the method of M-estimation as proposed in Iyer and Watson [8] to determine the set $T \supset R$ of points for computing the final local approximation. For solving the nonlinear system using the iteratively reweighted least squares method, use the coefficients of the linear interpolant obtained above as the initial point. Compute the final local approximation $P_k(x) = f_k + \sum_{j=1}^{m} a_j^{(k)} \left(x_j - x_j^{(k)} \right)$ using points $x^{(k)}$ and $x^{(i)}$, $i \in T$.

In the unlikely event that the Gramian matrix corresponding to the minimal set indexed by R is ill-conditioned, the entire set is rejected and the algorithm seeks to find the next-best minimal set. When the entire collection of s_k nodes is ill-conditioned, additional nodes are added by increasing N_p, up to a limit that is arbitrarily set at $N/2$ (half the data).

4. L2WPMA

L2WPMA (least squares weighted piecewise monotonic approximation, ACM TOMS Algorithm 863)[2] is Fortran software that calculates the best piecewise monotonic approximation to n univariate data points contaminated by random errors. A smooth function $f(x)$ is measured at the abscissae $x_1 < x_2 < \ldots < x_n$ and the measurements (function values) $(\phi_i \cong f(x_i) : i = 1, 2, \ldots, n)$ are assumed to contain random errors. It is also assumed that the number of sign changes observed in the sequence $(\phi_{i+1} - \phi_i : i = 1, 2, \ldots, n-1)$ is much greater than that number in the sequence $(f(x_{i+1}) - f(x_i) : i = 1, 2, \ldots, n-1)$. The problem, therefore, is reduced to the problem of calculating numbers $(y_i : i = 1, 2, \ldots, n)$ by making the "least" change to the data so that the sequence $(y_{i+1} - y_i : i = 1, 2, \ldots, n-1)$ has at most $k-1$ sign changes, where k is a small positive integer smaller than n. To achieve this, the vector y is calculated to minimize the expression

$$\phi(y) = \sum_{i=1}^{n} w_i (\phi_i - y_i)^2$$

subject to the constraints $(y_{t_j} \leq y_{t_j+1} \ldots \leq y_{t_{j+1}})$, when j is even and $(y_{t_j} \geq y_{t_j+1} \ldots \geq y_{t_{j+1}})$, when j is odd, where $(t_j : j = 1, 2, \ldots, k-1)$ are integers that satisfy the conditions

$$1 = t_0 \leq t_1 \leq \ldots \leq t_k = n,$$

and where the weights (w_i) satisfy the inequality $w_i > 0, i = 1, 2, \ldots, n$. The final constructed approximation is a continuous linear spline with breakpoints x_i, $i = 1, \ldots, n$, interpolating the points (x_i, y_i), consisting of k monotonic linear splines, alternately monotonically increasing and monotonically decreasing.

The value of k is provided by the user and can be selected by either inspecting the plotted data, or by forming tables of the first divided differences of the data and checking for sign alterations, or by increasing k until the differences $(y_i - \phi_i : i = 1, 2, \ldots, n)$ seem to be due to the errors of the data. The user is also required to specify if the first monotonic section of the data is increasing or decreasing. The variables $(t_j : j = 1, 2, \ldots, k-1)$ are the variables of the minimization calculation together with the components of y. This is a combinatorial problem. Demetriou and Powell [1991] have listed algorithms that solve this problem is fewer than $O(kn^2)$ computer operations. The dynamic programming technique used by L2WPMA divides the data into at most k disjoint sets of adjacent data and solves a $k = 1$ problem for each set. Thus, it efficiently calculates a global solution in only $O(n\sigma + k\sigma^2)$ computer operations, where $k \geq 3$ and σ is the number of local minima of the data.

The method is very efficient due to the following three properties. First, the calculation of the turning points (extrema) is restricted to local extrema. Second, as the data increases, the rightmost turning point index t_{k-1} is chosen to increase monotonically, which reduces the amount of computation as it shortens the range of the required approximation. Third, increasing the number of monotonic sections has very little effect on the overall numerical work and a monotonic approximation is calculated exactly once for a given section. The key property of the approximation is that it consists of optimal disjoint sections of monotonic components.

5. SAMPLE FUNCTION RESULTS

The performance of the linear Shepard method (LSHEP), robust linear Shepard method (RLSHEP), RIPPLE, and L2WPMA was compared using a piecewise linear function and a sine function. The plot for the piecewise linear function

$$f_1(x) = \begin{cases} (1+x), & \text{if } -1 \leq x < 0, \\ (1-x), & \text{if } 0 \leq x \leq 1, \end{cases}$$

is shown in Figure 1. The plots for the function $\sin(x)$ for $(-\pi \leq x \leq \pi)$ and $(0 \leq x \leq 10\pi)$ are shown in Figures 2 and 3, respectively. Each function was sampled at n_0 random points and contaminated with small random errors. The piecewise linear function shown in Figure 1 was contaminated further to contain $n_0/5$ outliers. The approximation error has been characterized using two error metrics. These are the maximum absolute error e_{max} and the root mean

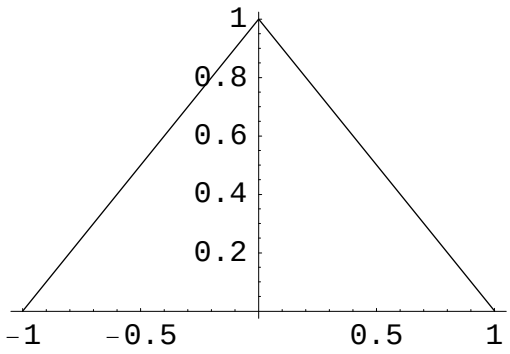

Figure 1. Piecewise Linear Function f_1.

TABLE 1. INTERPOLATION ERRORS FOR f_1.

	LSHEP		**RLSHEP**		**RIPPLE**	
n_0	e_r	e_{max}	e_r	e_{max}	e_r	e_{max}
10	9.81E-2	1.26E-1	7.26E-2	9.93E-2	2.55E-3	2.91E-3
20	9.63E-2	1.11E-1	6.12E-2	9.48E-2	2.42E-3	2.42E-3
30	9.46E-2	1.03E-1	6.04E-2	8.76E-2	2.33E-3	2.33E-3

TABLE 2. INTERPOLATION ERRORS FOR f_1.

	L2WPMA			
	$k = 2$**, I**		$k = 3$**, D**	
n_0	e_r	e_{max}	e_r	e_{max}
10	6.63E-2	9.87E-2	7.01E-2	9.94E-2
20	6.07E-2	8.93E-2	6.56E-2	9.01E-2
30	5.28E-2	7.63E-2	6.12E-2	8.76E-2

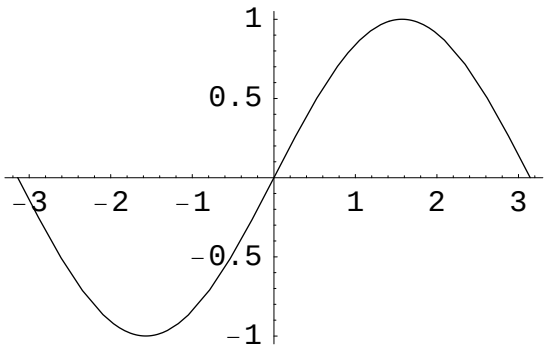

Figure 2. Sine Function, $-\pi \leq x \leq \pi$.

squared error e_r. The absolute approximation error is defined as

$$e_i = |\tilde{f}(x_i) - f(x_i)|,$$

where $\tilde{f}(x)$ is the approximating function, $f(x)$ is the test function, and x_i are 50 uniformly distributed points on the x-axis.

Using this notation, the maximum absolute error is

$$e_{max} = \max_{1 \leq i \leq n} e_i,$$

and the root mean squared error is defined as

$$e_r = \sqrt{\sum_{i=1}^{n} e_i^2 \Big/ n}.$$

The results are summarized in Tables 1–6. The results for L2WPMA are shown with different parameter values. k specifies the number of monotonic sections and the values I and D specify whether the first section is monotonically increasing or decreasing.

TABLE 3. INTERPOLATION ERRORS FOR $\sin(x)(-\pi \leq x \leq \pi)$.

	LSHEP		**RLSHEP**		**RIPPLE**	
n_0	e_r	e_{max}	e_r	e_{max}	e_r	e_{max}
50	9.65E-2	1.47E-1	5.68E-2	7.93E-2	7.93E-3	8.69E-3
75	9.57E-2	1.31E-1	5.65E-2	7.47E-2	7.91E-3	8.62E-3
100	8.85E-2	1.13E-1	5.67E-2	7.86E-2	7.85E-3	8.59E-3

TABLE 4. INTERPOLATION ERRORS FOR $\sin(x)(-\pi \leq x \leq \pi)$.

	L2WPMA					
	$k = 2$**, D**		$k = 3$**, D**		$k = 5$**, D**	
n_0	e_r	e_{max}	e_r	e_{max}	e_r	e_{max}
50	3.22E-1	5.56E-1	7.98E-3	1.56E-2	9.86E-3	2.03E-2
75	1.26E-1	4.48E-1	7.91E-3	1.52E-2	8.65E-3	1.72E-2
100	9.97E-2	5.23E-1	7.86E-3	1.52E-2	8.59E-3	1.63E-2

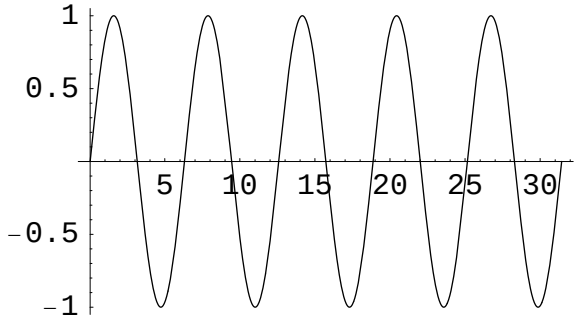

Figure 3. Sine Function $0 \leq x \leq 10\pi$.

TABLE 5. INTERPOLATION ERRORS FOR $\sin(x)(0 \le x \le 10\pi)$.

	LSHEP		**RLSHEP**		**RIPPLE**	
n_0	e_r	e_{max}	e_r	e_{max}	e_r	e_{max}
50	9.65E-2	2.15E-1	9.85E-2	1.70E-1	6.26E-3	7.29E-3
100	9.56E-2	1.92E-1	6.63E-2	9.72E-2	6.15E-3	6.98E-3
150	9.16E-2	1.73E-1	6.24E-2	9.63E-2	6.09E-3	6.50E-3
200	8.93E-2	1.51E-1	6.09E-2	9.86E-2	5.93E-3	6.46E-3

TABLE 6. INTERPOLATION ERRORS FOR $\sin(x)(0 \le x \le 10\pi)$.

	L2WPMA					
	$k = 5$**, I**		$k = 11$**, I**		$k = 13$**, I**	
n_0	e_r	e_{max}	e_r	e_{max}	e_r	e_{max}
50	4.65E-2	1.47E-1	5.29E-3	6.70E-3	6.26E-3	9.88E-3
100	4.22E-2	1.43E-1	4.93E-3	6.69E-3	6.15E-3	9.81E-3
150	3.96E-2	1.15E-1	4.63E-3	6.70E-3	5.99E-3	9.83E-3
200	3.83E-2	1.11E-1	4.42E-3	6.46E-3	5.91E-3	9.76E-3

6. CONCLUSION

As stated in Section 4, L2WPMA requires the user to provide two parameters along with the data values. The two parameters are the number of monotonic sections (k) and the specification whether the first section is increasing or decreasing. As seen in the results, if these parameters are not specified correctly, the performance of the method degrades. However, a good approximation is obtained when the user has prior information about the data.

The performance of RIPPLE is as good as the performance of L2WPMA (when the parameters required for L2WPMA are specified correctly). The primary advantage of RIPPLE is that the user is not required to have any prior knowledge of the data. Another advantage is that RIPPLE is a local method, which means that an approximation at a particular point can be obtained by considering only a subset of the data. This is particularly useful when the data sets are very large. Also, RIPPLE considers the impact of outliers while calculating the best approximation and downweights bad data points.

Like RIPPLE, L2WPMA is also suitable for large data sets as it never computes a monotonic approximation for a given data segment more than once. However, if the data contain random errors that are too small to cause many sign changes in the sequence of the first divided differences of the data, then the performance of the method may degrade. Also, the method does not have a scheme for giving bad data (outliers) smaller weights. This results in interpolation of data containing large errors, which can also have a significant impact on its performance. This is illustrated in the results shown in Table 2.

The performance of the linear Shepard algorithm is not good for piecewise linear functions as it "averages" all facets near a ridge. The robust linear Shepard method performs better than the standard linear Shepard method as it picks up the facet consistent with the majority of nearby data.

All results obtained are consistent with expectations and predictable from the nature of the approximation algorithms.

REFERENCES

[1] P. J. Besl, J. B. Birch, and L. T. Watson, "Robust window operators", *Machine Vision and Applications*, vol. 2, no. 4, pp. 179–191, 1989.

[2] I. C. Demetriou, "Algorithm 863: L2WPMA, a Fortran 77 package for weighted least-squares piecewise monotonic data approximation", *ACM Transactions on Mathematical Software*, vol. 33, no. 1, 2007.

[3] I. C. Demetriou and M. J. D. Powell, "Least squares smoothing of univariate data to achieve piecewise monotonicity", *IMA Journal of Numerical Analysis*, vol. 11, no. 3, pp. 411–432, 1991.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.

[5] R. Franke and G. Neilson, "Smooth interpolation of large sets of scattered data", *J. Numer. Methods Eng.*, vol. 15, pp. 1691–1704, 1980.

[6] V. Gantovnik, Z. Gürdal, and L. T. Watson, "Linear Shepard interpolation for high dimensional piecewise smooth functions", in *Proc. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf.*, CD-ROM, Albany, NY, 2004, 15 pages.

[7] P. Huber, *Robust Statistics*, John Wiley and Sons, New York, 1981.

[8] M. A. Iyer and L. T. Watson, "An interpolation method for high dimensional scattered data", in *Proc. 2006 Spring Simulation Multiconf., Business and Industry Symp.*, J. A. Hamilton, Jr., R. MacDonald, and M. J. Chinni (eds.), Soc. for Modeling and Simulation Internat., San Diego, CA, 2006, pp. 217–222.

[9] M. A. Iyer and L. T. Watson, "RIPPLE: Residual Initiated Polynomial-time Piecewise Linear Estimation", in *Proc. IEEE Southeastcon '07*, CD-ROM, Richmond, VA, pp. 444–449.

[10] R. Myers, "Classical and Modern Regression with Applications", Duxbury Press, 2nd edition, 2000.

[11] R. J. Renka, "Multivariate interpolation of large sets of scattered data", *ACM Transactions on Mathematical Software*, vol. 14, no. 2, pp. 139–148, 1988.

[12] P. J. Rousseeuw, "Least median of squares regression", *Journal of American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984.

[13] D. Shepard, "A two-dimensional interpolation function for irregularly spaced data", *Proceedings of the 1968 23rd ACM national conference*, pp. 517–523, 1968.