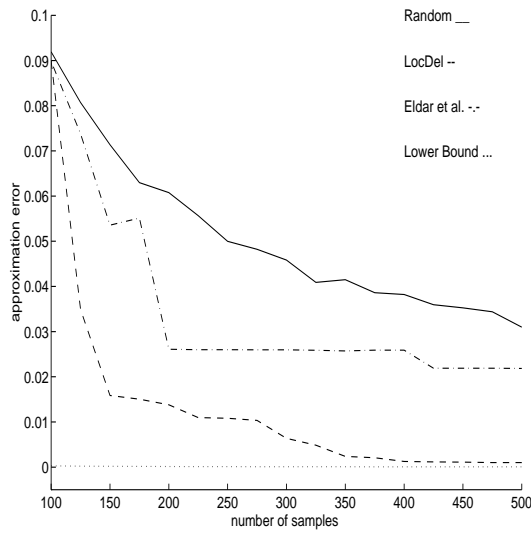


Figure 10: Triangulations and induced PLS's on a random 200 point sample of F_8 (a)-(b) Delaunay PLS (approximation error = .086) (c)-(d) ABN PLS (approximation error = .081) (e)-(f) LocDel PLS (approximation error = .045)

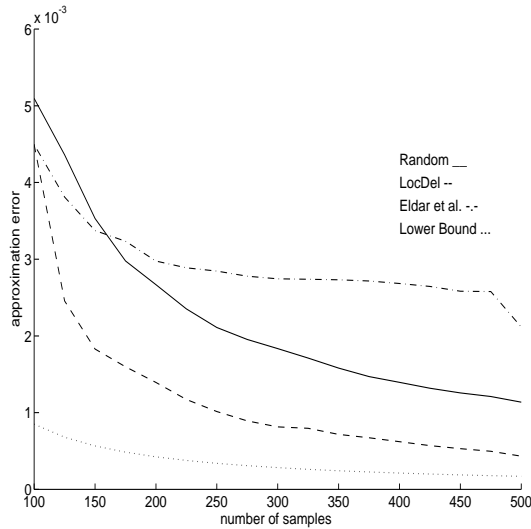
- SIAM J. Sci. Stat. Comput.*, 10(5):1063–1075, 1989.
- [5] N. Dyn, D. Levin, and S. Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, 10:137–154, 1990.
- [6] Y. Eldar. Irregular image sampling using the Voronoi diagram. Master’s thesis, Technion - Israel Institute of Technology, Faculty of Electrical Engineering, August 1990.
- [7] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Zeevi. The farthest-point strategy for progressive image sampling. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem*, volume 3, pages 93–97, 1994.
- [8] C.L. Lawson. Transforming triangulations. *Discrete Math.*, 3:365–372, 1972.
- [9] M. Margaliot and C. Gotsman. Piecewise-linear surface approximation from noisy scattered samples. In *Proceedings of Visualization '94*, pages 61–68. IEEE Computer Society Press, 1994.
- [10] E. Nadler. Piecewise linear best l_2 approximation on triangulations. In C.K. Chui, L.L. Schumaker, and J.D. Ward, editors, *Approximation Theory V*, pages 499–502. Academic Press, 1986.
- [11] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.
- [12] S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM J. Numer. Anal.*, 29(1):257–270, 1992.

whose computational complexity is usually irrelevant.

We have also proposed a method for adaptive sampling based on l_2 approximation by PLS considerations. It should be emphasized that the performance of an adaptive sampler depends heavily on the method used for surface reconstruction from the samples, hence it should be designed with this in mind. Ours relies on the fact that the reconstruction method is by PLS, hence it is not surprising that the results are good, even close to the best possible. The random sampler obviously is data-independent, and the sampler of Eldar et. al does not rely on any specific reconstruction method, so its performance is somewhat unpredictable.



(a)



(b)

Figure 9: Approximation errors of Loc-Del PLS’s on random and adaptive sample patterns of (a) F_8 and (b) F_4 .

References

- [1] Special issue on softcopy photogrammetric workstations. *Photogrammetric Engineering and Remote Sensing*, January 1992.
- [2] D. Cohen and C. Gotsman. Photorealistic terrain imaging and flight simulation. *IEEE Computer Graphics and Applications*, 14(2):10–12, 1994.
- [3] E.F. D’azevedo. Optimal triangular mesh generation by coordinate transformation. *SIAM J. Sci. Stat. Comput.*, 12(4):755–786, 1991.
- [4] E.F. D’azevedo and R.B. Simpson. On optimal interpolation triangle incidences.

compares the results to the asymptotic estimate (4) of Nadler, which is a lower bound on the approximation error. It can be seen that the freedom to choose the samples enables us to obtain an approximation significantly better than a typical random sample. The results are not optimal, but not far from it. For F_4 , the sample set and corresponding LocDel PLS of Eldar et. al performs much worse than the random sample set. This is because it concentrates samples in areas of large first derivative magnitude, hence misses the peak of the Gaussian. In contrast, the sample set generated by our adaptive sampler concentrates in areas of large *second* derivative magnitude, hence the corresponding LocDel PLS yields significantly better results.

4 Conclusion

We have proposed a method for the construction of a data-dependent triangulation of surface samples based on l_2 approximation error criteria. Our approach combines elements from known analytic results on quadratic functions with algorithmic optimization techniques producing data-dependent triangulations. This method differs from methods proposed in the past in that it does not attempt to minimize some heuristic PLS roughness measure. PLS's obtained using this method approximate the sampled function better, at the expense of a very small overhead in computing time. Even a larger overhead would be acceptable, as in most applications the PLS is constructed just once in a preprocessing stage,

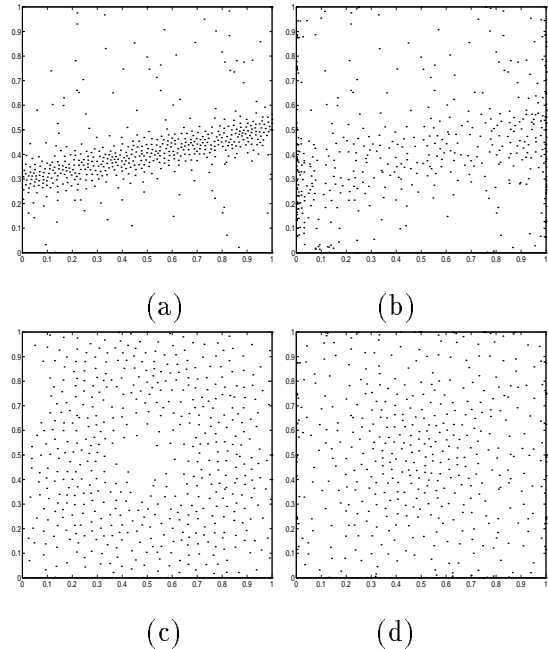


Figure 8: Sample patterns generated by adaptive samplers: (a) Eldar et. al on F_8 . (b) Ours on F_8 . (c) Eldar et. al on F_4 . (d) Ours on F_4 .

3.2 The Adaptive Sampler

Assume we have n samples: $\{(x_i, y_i, f(x_i, y_i))\}_{i=1}^n$ of some smooth unknown function f , and we wish to sample f again at a new point (x_{n+1}, y_{n+1}) . The question is what is the location (x_{n+1}, y_{n+1}) such that the new sample will be most useful (without knowledge of f). This obviously depends on the locations and values of all previous samples. An algorithm which generates such samples is called an *adaptive* sampler. An example of such an adaptive sampler was proposed by Eldar et al. [7, 6] for *image* sampling. This sampler prefers to sample first large areas which have not yet been sampled, or areas in which its estimate of the local function value variance is high. The latter is, loosely speaking, equivalent to areas where the magnitude of the first derivative is large. A good adaptive sampler should concentrate samples where they are needed most. The exact criterion depends on the method used for function reconstruction from the samples. We deal with the case where the reconstruction method is a PLS on a triangulation of the samples. The PLS we use is the one generated by the LocDel method described in Section 2. In this case, the most obvious candidate location for $n+1$ 'th sample is in the triangle having the greatest approximation error. We place the sample at the centroid of this triangle. To find this triangle we build the LocDel triangulation of the sample set, scan all triangles, build a weighted least square quadratic approximation over each one and calculate the approximation

```
/* Given a sample set S of size n,
 * produce one more sample location.
 */
Algorithm Sampler(S)
begin
  f' = LocDel-PLS(S)
  locate triangle T in f' with largest
  approximation error by building
  a quadratic approximation over
  each triangle.
  return centroid(T)
end;
```

Figure 7: Pseudo-code of the adaptive sampler

error (2). This contrasts with the quadratic approximation of Section 2, which was based on quadrilaterals. Here, the set V includes only the triangle vertices and its (between one and three) neighbors. If the number of points in V is less than six, we ignore the triangle. Pseudo code of our adaptive sampler appears in Fig. 7.

3.3 Experimental Results

We tested our adaptive sampling algorithm on F_8 and F_4 , starting with $n = 100$ random samples, applying our algorithm repeatedly to add a new sample until $n = 500$. To evaluate our results, we compared them to those of a simple random location sampler and the adaptive sampler of Eldar et al. The sample patterns generated by the two adaptive samplers are shown in Fig. 8. The approximation error of the LocDel PLS corresponding to the sample sets is shown in Fig. 9. Note that the case of the random sampler is essentially equivalent to the LocDel PLS constructed on the random inputs in Section 2.3. Here too the results are averaged over many possible random location sets. Fig. 9 also

3 Adaptive Sampling of Smooth Surfaces

3.1 Problem Description

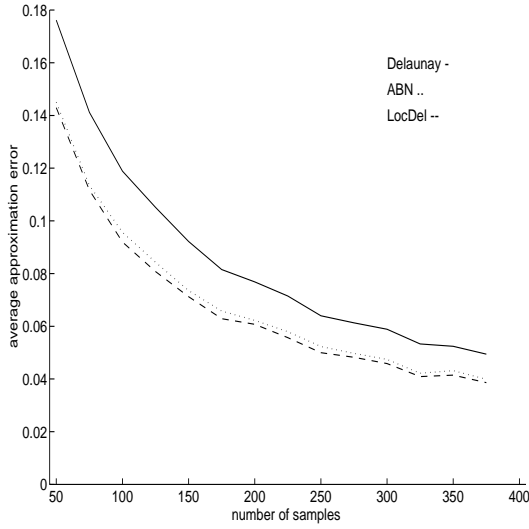
In this section we use the error expression (2) for *adaptive sampling*. In the previous section we assumed the sample set was given, therefore the only freedom in constructing the approximating PLS is choosing the triangulation of the sample set. In this section we assume that the algorithm may also *choose* the sample set, therefore better able to optimize the PLS approximation for a given *number* of samples. Nadler [10] obtained an asymptotic estimate for the l_2 approximation error of a function $f \in C^3(\Omega)$ by the best PLS f' defined on n (optimal) samples of f on the domain Ω :

$$\lim_{n \rightarrow \infty} n \cdot \|f - f'\|_2 = \left[\int \int_{\Omega} J(x, y)^{\frac{1}{3}} dx dy \right]^{\frac{3}{2}} \quad (4)$$

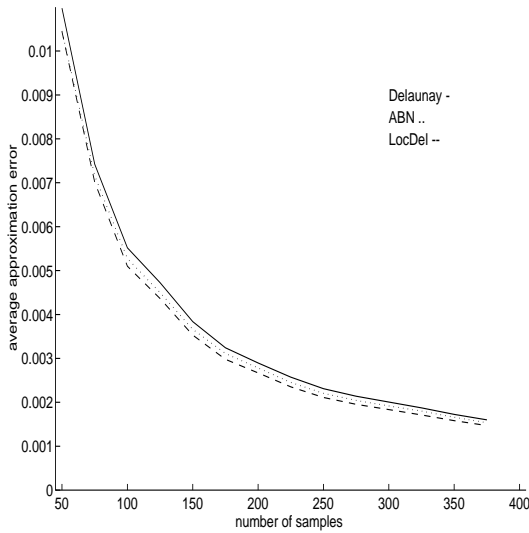
where

$$J(x, y) = \begin{cases} \frac{1}{180} \det H(x, y) & \det H(x, y) > 0 \\ \frac{1}{225} |\det H(x, y)| & \det H(x, y) < 0 \end{cases}$$

and $H(x, y)$ is the Hessian matrix of f . This means that for a large number of samples n , the smallest error obtainable is $\theta(1/n)$. How to construct this optimal sample set and its optimal PLS is an open question. We are not able to provide such a construction, but do provide an incremental algorithm for sample generation based on arguments similar to those used in the previous section.



(a)



(b)

Figure 6: Approximation errors of various PLS's on random samples of (a) F_8 and (b) F_4 .

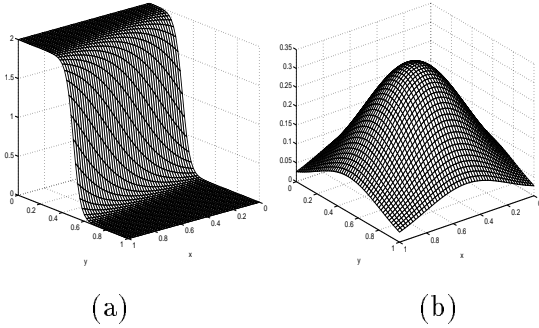


Figure 5: Test functions (a) F_8 and (b) F_4 .

and compare them with those of [5], we constructed three PLS's. The first was obtained using the data-independent Delaunay triangulation (referred to as the *Delaunay PLS*). The second was obtained by applying the data-dependent ABN LOP procedure described in [5] (referred to as the *ABN PLS*). The third was obtained by applying our algorithm (referred to as the *LocDel PLS*). For each resulting PLS we computed the l_2 approximation error relative to the sampled function by numerical integration. In order to test the performance of the data-dependent triangulation methods as a function of n - the sample set size, they were applied to many random sample sets of n points, and statistics gathered. For each n , we ran the triangulation algorithms on 100 sample sets, obtaining three l_2 -approximation error vectors of length 100:

$$\mathbf{e} = (\underline{\epsilon}_{DelaunayPLS}, \underline{\epsilon}_{ABNPLS}, \underline{\epsilon}_{LocDelPLS})$$

Fig. 6 shows the means of these vectors, as a function of n . The results indicate that both the ABN PLS and LocDel PLS achieve a better approximation than the Delaunay PLS, with the

LocDel PLS providing the better approximation between the two. For F_8 , which is a function with a clearly preferred direction, both the ABN PLS and LocDel PLS are of similar quality, significantly improving the Delaunay PLS. For F_4 , ABN PLS reduces the error, relative to the Delaunay PLS, by about 4%, and, the LocDel PLS by about 9%. Note that this is not true when n is small, probably because, in this case, the local quadratic approximations are inaccurate. More information on the algorithms' relative performance may be obtained by inspection of the 3×3 correlation matrix of \mathbf{e} . For example, for F_8 with $n = 300$:

$$\text{corr}(\mathbf{e}) = \begin{pmatrix} 1.0000 & 0.9071 & 0.8764 \\ 0.9071 & 1.0000 & 0.9404 \\ 0.8764 & 0.9404 & 1.0000 \end{pmatrix}$$

The values in this correlation matrix, which are all close to 1, indicate that the average improvement obtained by the LocDel PLS algorithm is a consistent phenomenon among the majority of random inputs.

Figure 10 shows the resulting PLS's approximating F_8 for a random sample of 150 points. The ABN PLS prefers triangles which are long in the direction of minimal surface curvature. The LocDel PLS takes this further, producing more of these triangles, resulting in a closer approximation.

global functional on the PLS (such as $R(f')$) but local error estimates instead. However, in all our numerical experiments the algorithm reaches a “steady state” where a very small number of edge swaps is performed during each scan of the triangulation. When this state is reached we terminate.

The weighted least squares method we use to construct the local quadratic approximation g to the unknown f proceeds as follows:

Calculate the six independent entries of the symmetric real matrix H_Q that minimizes

$$\sum_{Q_i \in V} \frac{1}{w_i^2} [(x_i \ y_i \ 1) H_Q (x_i \ y_i \ 1)^t - z_i]^2$$

The summation is over the set V which includes the four vertices of the quadrilateral Q and its two to four neighbors in the current triangulation, a total of six to eight points (see Fig. 3).

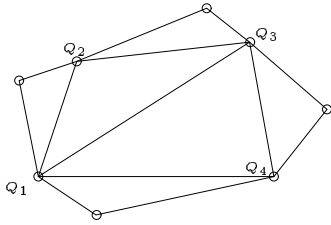


Figure 3: The vertex set V used for constructing the quadratic approximation over the quadrilateral $Q = Q_1Q_2Q_3Q_4$.

Each vertex is weighted by its distance to the quadrilateral centroid C_Q :

$$w_i = \|Q_i - C_Q\|_2 \quad ; \quad C_Q = \frac{Q_1 + Q_2 + Q_3 + Q_4}{4}$$

Fig. 4 summarizes our data-dependent triangulation algorithm, which we call the LocDel-PLS procedure, in pseudo-code.

```

/* Produce PLS f' given
 * sample set S of size n.
 */
Algorithm LocDel-PLS(S)
begin
  TR := Delaunay(S);
  while not (steady-state) do
    foreach convex quadrilateral Q in TR do
      begin
        Compute least-squares quadratic form
        g over Q and neighboring
        vertices:
        Calculate the square errors for
        the two possible triangulations of Q;
        Choose the triangulation with
        lower square error;
      end;
    return TR;
  end;

```

Figure 4: Pseudo-code of the LocDel data-dependent PLS generator.

2.3 Experimental Results

We tested our LocDel-PLS algorithm on samples of the test functions $F_i : [0, 1]^2 \rightarrow \mathbb{R}$ used in [5]. Of particular interest are:

$$F_8(x, y) = \tanh(-3g(x, y)) + 1$$

where :

$$g(x, y) = 0.595576(y + 3.79762)^2 - x - 10$$

and

$$F_4(x, y) = \frac{1}{3} \exp \left[\frac{-81}{16} \left[\left(x - \frac{1}{2} \right)^2 + \left(y - \frac{1}{2} \right)^2 \right] \right]$$

The function F_8 simulates a sharp rise, whose contour lines are the parabolas $g(x, y) = \text{const}$, and F_4 is a gentle Gaussian hill. (see Fig. 5). The functions were sampled on random points distributed uniformly in $[0, 1]^2$. These samples served as input to our algorithm which produced a PLS f' as output. To evaluate our results,

ness measure of a PLS and try to minimize it. Let $T_1 = \{p_1, p_2, p_3\}$ and $T_2 = \{p_2, p_3, p_4\}$ be two triangles of the PLS f' , with common edge $\overline{p_2p_3}$, and n_i be their vector normals (see Fig. 1). Define $ABN(e)$ of edge $e = \overline{p_2p_3}$ to be the Angle Between the Normals n_1 and n_2 .

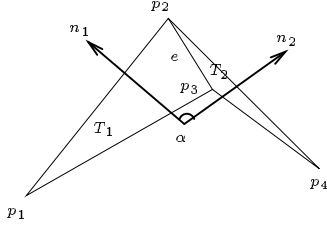


Figure 1: Two triangles of a PLS and their normals.

A possible PLS roughness measure is:

$$R(f') = \sum_{e \in \{\text{edges of } f'\}} ABN(e) \quad (3)$$

and the optimal triangulation would be the one inducing the PLS f' minimizing $R(f')$. To minimize $R(f')$, Dyn et. al used Lawson’s optimization procedure (LOP) [8]: Starting from the Delaunay triangulation, every edge of the triangulation, which is a diagonal of a convex quadrilateral Q , is replaced by the other diagonal of the quadrilateral (replacing the two triangles T_1, T_2 by two others T_3, T_4) if this decreases R (see Fig. 2). The procedure is repeated until no more edge swaps are required in the entire triangulation. This method was recently extended [9] to the case of noisy samples, as is frequently the case in real applications.

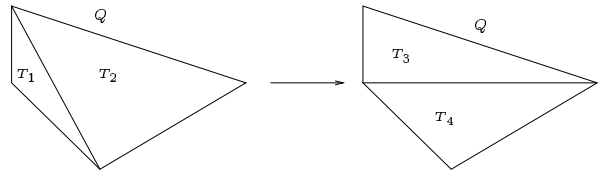


Figure 2: Swapping diagonals of a convex quadrilateral.

2.2 The LocDel PLS

We propose to obtain an optimal triangulation of a given sample set using LOP based on l_2 -approximation criteria. This is a more direct way of obtaining an optimal triangulation, since, ultimately, we want to minimize the l_2 approximation error between the PLS and the sampled, albeit, unknown, function. We will also, somewhat implicitly, assume that the sampled function is smooth, so that approximating it locally by a quadratic is valid. Given a convex quadrilateral Q in a triangulation of the sample points, we first construct a quadratic $g(x, y)$ over Q , by weighted least squares, which hopefully approximates the unknown f well in the vicinity of Q . Then, denoting the approximation errors on the four possible triangles T_i by $e_i = e(T_i, g)$ $i = 1, \dots, 4$ (see Fig. 2), the edges are swapped if $e_1^2 + e_2^2 > e_3^2 + e_4^2$. Note that if g is convex, this criterion is equivalent to performing a “local” Delaunay triangulation of just Q , after performing a coordinate transformation A on the vertices of Q , such that $H_Q = 2A^T A$, where H_Q is the constant Hessian of g .

It is important to note that the LOP may not converge because we are not minimizing any

to construct a PLS f' best approximating (the unknown) f , induced by a triangulation of the projected set: $V = \{(x_i, y_i)\}_{i=1}^n$. Since the triangulation of V completely determines the PLS f' , the only question to be answered is which, among the many possible, is a good triangulation ?

Until recently the Delaunay triangulation of V was considered optimal because it tends to avoid long and thin (skinny) triangles [11]. This triangulation is independent of the sample values z_i . A series of works studying the problem of the approximation of a *known* function f by a PLS have shown that a Delaunay triangulation of the sample set is suitable only for functions whose second derivatives do not exhibit extreme behavior in any specific orientation. In cases where this is not true, long and thin triangles might actually be desirable. Nadler [10] was the first to investigate the optimal *shape* for an l_2 -approximating triangle of fixed area. His results indicate, loosely speaking, that the triangle should be long in the direction of minimal second derivative of the approximated function f . This is consistent with the expression in (2). Triangulations depending on f or its samples are called *data-dependent* triangulations. Rippa [12] dealt with the construction of a data-dependent triangulation for optimal l_2 -approximation of a known *convex quadratic* function. D'azevedo [4] studied this topic for l_∞ -approximation. They both concluded:

- For $f_c(x, y) = x^2 + y^2$, the optimal triangulation is the Delaunay triangulation.

- For a general convex quadratic f , the optimal triangulation is obtained by applying an affine coordinate transformation A , such that $H = 2A^T A$ (where H is the Hessian matrix of f), on V , performing Delaunay triangulation on the *transformed* locations, and then applying the inverse affine transformation.

An intuitive justification of these conclusions is the following: For $f_c(x, y) = x^2 + y^2$ we would like to use equilateral triangles. When the points are scattered, the Delaunay triangulation gives the most “equilateral” triangles. For a general convex quadratic $f(x, y)$, we apply a coordinate transformation $A : (x, y) \rightarrow (u, v)$ so that $f(u, v) = u^2 + v^2$ and then apply the Delaunay triangulation in the (u, v) plane. D'azevedo [3] further showed how the latter method may be generalized to arbitrary (known) functions. This general case requires a coordinate transformation considerably more complex than a simple affine one.

The problem we deal with is complicated by the fact that it is ill-posed, i.e. since the function to be approximated is unknown (therefore also the exact behavior of its derivatives), and many possible triangulations, each inducing a different PLS, exist, there is no real reason a priori to prefer one triangulation over another. One possible heuristic is to assume the sampled function was smooth, suggesting that we should prefer a “smoother” PLS, or, equivalently, reject “rough” PLS's. Dyn et. al [5] define a rough-

In some applications, the surface sample set is not given, and it is up to the “user” to decide where to sample the surface. This leaves more room for optimization of the samples, i.e. placing them where they are most beneficial. In this case, a better approximation of the surface by constructing a PLS on n carefully chosen sample points is possible, as opposed to a PLS constructed on n arbitrary given points.

In this paper we deal both with the problem of constructing an optimal PLS on a *given* sample set, and with the problem of constructing an *optimal* sample set and PLS based on it.

1.1 Local Approximation of Quadratic Functions

We begin our analysis by studying the relatively simple class of quadratic functions. Let $f(x, y)$ be the quadratic function:

$$f(x, y) = \frac{1}{2}(x \ y \ 1)H(x \ y \ 1)^t, \quad (1)$$

where H is a constant 3×3 real symmetric matrix, and let Δ be a 2D triangle defined by three non colinear vertices: $\{v_i = (x_i, y_i)\}_{i=1}^3$. The three points: $\{(x_i, y_i, z_i = f(x_i, y_i))\}_{i=1}^3$ define a triangle T in space, coinciding with the surface $z = f(x, y)$ at its vertices. Define the l_2 approximation error $e(\Delta, f)$ of f by T as:

$$e(\Delta, f)^2 = \int \int_{\Delta} [f(x, y) - T(x, y)]^2 dx dy$$

(l_p approximations are defined analogously using the l_p norm). Nadler [10] obtained the following

analytic expression for this error:

$$e(\Delta, f)^2 = \frac{Area(\Delta)}{180} ((d_1 + d_2 + d_3)^2 + d_1^2 + d_2^2 + d_3^2) \quad (2)$$

where $d_i = \frac{1}{2}(v_{i+1} - v_i)H'(v_{i+1} - v_i)^t$ ($v_4 \equiv v_1$), such that H' is the upper left 2×2 sub-matrix of H . This immediately implies the following:

1. $e(\Delta, f)$ is independent of the linear and constant terms of f .
2. Let $f_c(x, y) = x^2 + y^2$. Among all triangles Δ with equal area, $e(\Delta, f_c)$ is minimized by the equilateral triangle. Indeed, for f_c we have $H = 2I$, hence $d_i = \|v_{i+1} - v_i\|^2$, i.e. the square of the side length. The symmetry in (2) then implies that minimal error is obtained when all d_i 's are equal.

Equation (2) can help us estimate the quality of triangulation-based piecewise-linear approximations of arbitrary smooth functions. Naturally, we cannot assume that an arbitrary function is quadratic, but, if it is smooth enough, it can be approximated locally, in the area of each individual triangle, as a quadratic function.

2 Data Dependent Triangulations

2.1 Problem Description

In this section we demonstrate the first use of (2). Let $z = f(x, y)$ be some unknown smooth surface and assume we are given a set of n samples of f : $S = \{(x_i, y_i, z_i = f(x_i, y_i))\}_{i=1}^n$. We would like

Approximation of Smooth Surfaces and Adaptive Sampling by Piecewise-Linear Interpolants

Michael Margaliot

Craig Gotsman

Dept. of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel

Dept. of Computer Science
Technion - Israel Institute of Technology
Haifa 32000, Israel

Abstract

Let $f(x, y)$ be a quadratic function and $T(x, y)$ a 3D triangle coinciding with the surface $z = f(x, y)$ at its three vertices. We use an analytic expression for the l_2 approximation error of T relative to f :

$$\|f - T\|^2 = \int \int_{\Delta} [f(x, y) - T(x, y)]^2 dx dy$$

where Δ is the projection of T onto the (x, y) plane, in two applications: Piecewise-linear approximation of smooth surfaces by data dependent triangulations of a sample set, and adaptive sampling of smooth surfaces. These applications are important in geometric surface modeling and visualization. The performance of our algorithms is shown to be better than existing methods dealing with these problems, at the price of some minor computation overhead.

Keywords: Surface, Adaptive Sampling, Piecewise-Linear, Approximation, Data-Dependent Triangulation.

1 Introduction

The general problem of approximating a smooth 2D function by a piecewise-linear surface (PLS) arises in a variety of applications where surface samples are either given, or obtainable at will. For example, the reconstruction of terrain surfaces from random digital terrain models (DTM's) extracted by automatic methods, such as matching stereo image pairs (see the many articles on this subject in [1]). These methods obtain terrain elevation samples wherever possible, usually at feature points, resulting in a data set consisting of points at essentially random locations in the plane.

The reason the approximation is done with a PLS, namely, a collection of triangles, is that it is the simplest method possible. Moreover, triangles are standard geometric primitives in modern graphics engine hardware. Terrain visualization by texture-mapped aerial imagery onto surface triangles is a popular graphics application in visual simulation environments [2].