Reproducing Kernels for Visual Surface Interpolation.

CUCS-186-85

by

Terrance E. Boult[†]

Columbia University, Department of Computer Science

§0 Abstract.

We examine the details of two related methods for the recovery of visual surfaces from sparce depth data. The methods use the reproducing kernels of Hilbert spaces to construct a spline interpolating the data, such that this spline is of minimal norm. We discuss the numerical properties of the two methods presented, and give example interpolations.

[†] This work supported in part by DARPA grant N00039-84-C-0165 and in part by NSF grant MCS-782-3676.

§1 Introduction.

It has been shown that when presented with sparce depth data (say from random dot stereo grams) that the human visual system infers smooth surfaces passing through these data points. [Grimson 81] presented a computational model of this process in the human visual system, and suggested an algorithm that could be used to recover the perceived surface from the depth data. The idea of visual surface interpolation is to take a sparce set of depth values, and from them, calculate the surface passing through these points that seems to model the surface that humans infer from those same data points.

Although it may be fruitful, from a psychological point of view, to develop algorithms that are physically realizable, it may also increase the computational complexity of the algorithms developed. Therefore, in this paper we examine two related methods for the solution of this visual surface interpolation problem, that may, under the right circumstances, yield more efficient algorithms. We shall refer to these methods as the methods of reproducing kernels. (The term reproducing kernel comes from certain properties of the functions that are not important for the developments in this paper.) The methods use the reproducing kernels of Hilbert and semi-Hilbert spaces to calulate splines of minimal norm. The use of surfaces of minimal norm as the visual surface interpolating the depth data is done in spirit of the minimization approach used in [Grimson 81]. These methods can be used under a wider range of assumptions than the method pioneered by Grimson, and also have some representational advantages over the grid of depth values that were used in [Grimson 81].

In section 2 we derive an precise formulation of the visual surface interpolation problem. In section 3 we present the general form of two methods for the solution of this problem. In section 4 we examine the assumption on the information we are going to allow in the recovery of these surfaces. Section 5 is a discussion of the advantages of splines in functional form as a representation, and the pros and cons (including time and space complexity issues) of using reproducing kernels to calculate the splines. In section 6 we present in detail the two methods as applied to a particular visual surface reconstruction problem. Section 7 gives our conclusions and discusses future work on this topic.

§2 The Problem.

The naive formulation of the problem to be solved is :

> to find "the best approximation" to any given physical surface using only the knowledge of a number of points thereon, where we require the surface to be interpolatory, i.e. to pass through all the given data.

A major difficulty with this formulation of the problem is that the problem is not well posed, inasmuch as the given information does not uniquely determine the solution. In fact, given any set (of zero measure) of points on a surface there are infinitely many surfaces interpolating those points. To alleviate this problem, we must somehow restrict the class of allowed surfaces and/or give some method of ranking the "plausibility" of a surface.

Approximation theorists have developed methods of insuring that an interpolation problem has a unique solution. One of the classical methods (applied to visual surface interpolation in [Grimson 81] and [Kender, Lee, and Boult 85]) is to use a functional on the surface as a measure of the "unreasonableness" of the surface, and to restrict the allowed class of surfaces to make this functional a norm or semi-norm on the space of allowed surfaces. This type of formulation insures that there exists a unique solution to the problem of finding a surface from the allowed class which minimizes the functional (and hence is the most reasonable). Throughout this paper we shall *assume* that this type of formulation is appropriate for the problem of visual surface interpolation. We shall not investigate which classes of surfaces are most appropriate, nor which functionals may be good measures of the unreasonableness of a surface.

In what follows we choose to define "best approximation" in terms on minimal error. We *assume* that error can measured by a norm with respect to the given class of functions (The interpolating surface is assumed to be approximating a surface from the given class.). The norm might be the **sup** norm (i.e. the maximal difference between the actual surface and the approximation), or the L^2 norm (integral of the square of the difference at each point). The error may be measured in either a relative (e.g. error of 5%) or an absolute sense (e.g. the surfaces never differ by more than .1 mm) depending on the goals of the user.

Combining these assumptions the formulation of the problem of visual surface interpolation from sparse depth data becomes :

> Let F_1 be the space of allowed surfaces. Let F_2 be the elements of F_1 restricted to some finite domain D (we shall only be interested in recovering a finite portion of a possibly infinite surface). Let $\Theta(f)$: $F_1 \rightarrow \Re$, be a functional measuring the "unreasonableness" of a surface (i.e. the more reasonable a surface f, the smaller $\Theta(f)$). Let $N(f) \equiv \{z_1,\ldots,z_k\} \equiv \{f(x_1,y_1),\ldots,f(x_k,y_k)\}$ be the allowed information (i.e. the allowed input to solve the problem is k depth values of the surface.) Then the visual surface interpolation problem is to find $f^* \in F_2$, (using only information N(f)) such that $\Theta(f^*) = \min_{g \in F_1} \Theta(g)$.

[Kender, Lee and Boult 85] show (as a special case of work on information based complexity [Traub and Wozniakowski 80], [Traub, Wasilkowski and Wozniakowski 83]) that given above formulation the surface minimizing the functional $\Theta(f)$ will also be the minimal error surface with respect to the class F_2 for almost *any error* norm.

§3 The Method of Reproducing Kernels.

The method of Reproducing kernels calculates a spline function that *exactly* solves the continuous problem of finding the function from the class F_1 that minimizes $\Theta(f)$.

For these methods to be appropriate it is sufficient to have F_1 be a semi-Hilbert space and $\Theta(f)$ the associated semi-norm with null space $\prod_{m-1}$. (Throughout this paper $\prod_{m-1}$ is the space of bivariate polynomial of degree m-1). To insure uniqueness of the solution we must assume that the information $N_k(f)$ contains a $\prod_{m-1}$ unisolvent subset, i.e. there exists a set J (a subset of the index set $I = 1 \ldots k$) of indices and a set of information points $\{x_j,y_j\}_{j \in J}$ and associated information values z_j such that there exists a unique $p_j(x,y) \in \prod_{m-1}$ satisfying the condition $p_j(x_j,y_j) = z_j \ \forall\, j \in J$.

In what follows we examine two separate reproducing kernel based representations of the solution to this minimization problem, the first pioneered by [Duchon 76, 77] the other by [Meinguet 79a, 79b, 83]. The advantages and disadvantages of each method shall follow the general description of both methods. Later sections of this paper shall give detailed examples of each method.

§3.1 Duchon's Method.

Duchon, extending the work of [Atteia 76] to the case of semi-Hilbert spaces, noted that the solution to the problem of finding an interpolating function of minimal norm in the semi-Hilbert setting could be written down in terms of K((x,y);(s,t)), the reproducing kernel of F_1. (These reproducing kernels need not always exist, and even when they do it may be very difficult to derive a closed form representation. However, for a number of "possibly appropriate" classes they are known in closed form.)

Given a reproducing kernel K((x,y); (s,t)) for F_1 we can write the interpolatory spline that minimizes $\Theta(f)$ as

$$\sigma_D(x,y) = \sum_{i=1}^{k} \alpha_i \cdot K((x,y); (x_i,y_i)) + \sum_{i=1}^{d} \beta_i \cdot q_i(x,y) \tag{3.1}$$

where $\{q_i\}_1{}^d$ (d = cardinality of the set J) is a basis for $\prod_{m-1}$, the null space of $\Theta(f)$. The coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ of the interpolating spline can be determined from the solution of a (k+d) by (k+d) dense linear system.

Recalling that $N(f) \equiv \{z_1,\ldots,z_k\} \equiv \{f(x_1,y_1),\ldots,f(x_k,y_k)\}$, where (x_i,y_i) are the location of the function (depth) values we can express this linear system as follows:

$$\sum_{i=1}^{k} \alpha_i \cdot K((x_j,y_j); (x_i,y_i)) + \sum_{i=1}^{d} \beta_i \cdot q_i(x_j,y_j) = z_j, \qquad j = 1, \ldots, k$$

(3.2)

$$\sum_{i=1}^{k} \alpha_i \cdot q_i(x_j,y_j) = 0, \qquad j = 1, \ldots d.$$

Duchon shows that this representation yields a σ_D that minimizes the functional $\Theta(\sigma_D)$ and is unique if the set $z_i = f(x_i,y_i)$, $i = 1..k$ contains a $\prod_{m-1}$ unisolvent subset.

§3.2 Meinguet's Method.

Another representation, due to Meinguet, uses that fact that we can separate the space F_1 into $X_0 \oplus \prod_{m-1}$, where $\prod_{m-1}$ is the null space of $\Theta(\cdot)$, $X_0 \equiv \{g \in F_1: g(x_j,y_j) = 0, \forall j \in J\}$ and $\oplus$ is a (topological) direct sum. (Recall that J is the set of indices of the $\prod_{m-1}$ unisolvent subset of the information.) Then X_0 is a Hilbert space with norm (not semi-norm) $\Theta(\cdot)$. Then given the reproducing kernel $K_M(s,t; x,y)$ of X_0 (Which can be expressed in terms of the reproducing kernel $K((s,t);(x,y))$ of F_1 and the functions $q_j(x,y)$) the interpolating spline is given by :

(3.3)
$$\sigma_M(x,y) = \sum_{i \in I-J} \gamma_i \cdot K_M((x,y); (x_i,y_i)) + \sum_{j \in J} z_j \cdot q_j(x_j,y_j)$$

where the coefficients γ_i can be calculated from the (k-d) by (k-d) (d equals the cardinality of the set J) dense linear system given by:

(3.4)
$$\sum_{i \in I-J} \gamma_i \cdot K_M((x_k,y_k); (x_i,y_i)) = z_k - \sum_{j \in J} z_j \cdot q_j(x_j,y_j), \ \forall k \in I-J.$$

§3.3 Discussion of the Two Methods.

The major advantage of the representation given by (3.3) (hereafter referred to as the σ_M representation) is that the system defined by (3.4) is always symmetric and positive definite. This is a important property from the numerical analytic point of view, insuring the numerical stability of certain special algorithms for the solution of the system. The biggest drawbacks to the σ_M representation are the complexity (and therefore the time required for calculation) of the generating kernel (as compared to the kernel functions for the σ_D representation); and, the need to explicitly calculate a unisolvent set of data, and the functions $p_j(x,y)$ interpolating that unisolvent set of data.

The major advantages of the representation given by (3.1) (hereafter referred to as the σ_D representation) is the symmetry of the system, and the simplicity (relative to the kernels of the σ_M representation) of the reproducing kernels. The major disadvantage of the representation is that the system generated by (3.2) is an indefinite system. (If fact the system will always have d negative eigenvalues, where d is the cardinality of the set J). This limits the algortithms that can be used for its solution, and also adversly effects the time complexity of some of the remaining algorithms.

We end with a note that although equations (3.1) and (3.3) seem different, if the class of functions, and the norm are the same, then the resulting splines are exactly the same function! This follows directly from the uniqueness of the function minimizing the functional $\Theta(\cdot)$ in the class F_2.

§4 Allowed Information.

We have not really discussed what type of information is available to the algorithm for the solution of the problem of the reconstruction of visual surfaces. The human visual system definitely has more information than pure depth data, so we must explicitly condider what information to allow our computer algorithms. For simplicity we shall assume that the only information about the surface is the function values (depth values) at the k points (x_i,y_i), $i = 1..k$. Note that is not a restriction of the reproducing kernel methods, in fact they are easily extensible with respect to the allowed information. (As long as the information is linear, e.g. derivatives or integrals of the surface. Linearity however rules out someimportant visual clues e.g. texture, most shading and all perspective clues).

The complexity of the calculation of the parameters of the splines σ_D and σ_M can be substantially lowered if we fix the location of the information samples. That is for all surfaces, we always sample the data at fixed locations. It is particularly convenient if these data points form a regular grid. Through out the rest of this paper we shall consider three separate cases: sample locations fixed on a regular grid; sample locations fixed but not on a regular grid; and sample locations that depend on the surface being approximated.

Fixing the location is not a sever limitation, compared with the information used in other algorithms, in terms of the error of the approximation we can derive. It is a limitation on the methods which can be employed in the gathering of data. In particular it rules out the use of a stereo based algorithm to generate the depth data, but lends itself well to 'industrial uses' with depth data from laser rangefinders or projection-triangulation based methods.

We digress from the reproducing kernel methods to note that although Grimson proves what he calles the surface consistency constraint (no news is good news) which relates the probability of a zero crossing in the Laplacian of the Gaussian (∇^2G) of the image to the variation of the surface, his algorithms (nor any other known to this author) make use of this "extra information" about the relationship of the zero crossings and the variation of the surface. Instead, Grimson choose to minimize the total surface variation, not the variation between individual zero crossings. Note that this implies even if we fix the location of our depth values on a regular grid, it is still at least equal in power to the information used by other algorithms recovering surfaces through minimization of functionals.

§5 Some Pros and Cons of Reproducing Kernel Splines as a Representation of Visual Surfaces.

Both of the reproducing kernel methods presented in section 3, represent the final interplatory surface as a spline function, in particular a weighted sum of translated reproducing kernels. We first discuss the advantages and disadvantages of the choice of a spline (in a functional form) as the representation. Then we discuss the difficulties of the use of reproducing kernels as the means of calculating this spline.

Some of the advantages of the spline representation are:

It gives us a functional form for the surface. In particular, we can then symbolically compute functionals on the surface, and using results from the General Theory of Optimal algorithms, see [Traub, Wozniakowski 80] and [Traub, Wasilkowski, and Wozniakowski 83] we know that the optimal error approximation to any linear functional of the surface is the desired functional applied to the spline. For example, if we wished to approximate the first derivatives of the surface at a number of points, we would simply calculate the spline as above (either σ_D or σ_M) and then the minimal error approximations would be given by $\partial\sigma/\partial x$ and $\partial\sigma/\partial y$. We could just as easily calculate any derivative (as long as the class F_1 was sufficiently smooth for the derivative to exist) or integral of the surface.

Because we have a functional form for the surface, we can recover the value of the surface at any point, and are not limited to points on a regular grid, or points in some predefined domain (though the error of the approximation grows larger with the distance to the nearest information point).

If the number of information points is small compared to the number of recovery points, the the spline representation can even be used as an efficient means of storage (on the computer) for the surface. The number of operations necessary to recover the surface (given that the k parameters are known) is O(n*k) where n is the number of recovery points and k is the number of information points. Furthermore, the calculations can all be done in parallel in a straight forward SIMD (a type of parallel algorithm) fashion.

The splines are the provably optimal error estimation (unique) for any surface from the class fitting the given data, see [Kender, Lee, and Boult 85]

The splines can in general be incrementally updated with respect to amount of information i.e. adding 1 new point does not requires the solution of a new system of equations :

Some of the disadvantages of a spline representation are:

If the number of information samples is high it can require tremendous amounts of storage (as compared with. say generalized cylinders) to store the information necessary to recover the surface. Recovery requires 3k units of storage (the location of the k information samples which takes 2k numbers and the approximately k splines coefficients for the spline itself). The storage for the system of equations that yield the coefficients of the spline requires $\approx k^2/2$;

The actual parameters of the spline depends heavily on the location of and value of every information point therefore error in any information sample will contaminate the entire spline surface; This also implies that if one data value is changed, every surface value must also be updated ;

The time complexity of the reproducing kernel methods for the calculation of the spline depends on the information. Given that we have the coefficients of the spline, the time complexity of recovery is linear in the number of information points.

If the location of the information is allowed to vary from reconstruction to reconstruction, the the time complexity of the current methods is $O(k^3)$ and comes from the direct solution to one of the linear systems (3.2) or (3.4).

If the location is fixed, then we can precomputing the coefficients of the basis splines, which involves the solution of the system (let us call it A) for k basis vectors or simply the inversion of the system. Total precomputation cost $\approx k^3$. From the inverse of the system we can recover the coefficients of the spline for any given set of data, z by taking A^{-1} z. Then the cost to recover the coefficients of the spline is $\approx k^2$.

One of the many topics not addresses in this exposition is the choice of the appropriate class, and the norm for that class. If the class belongs to one for which the reproducing kernel is already known (for a partial list see [Boult 85]), the algorithms can be directly applied, with the only change being the introduction of a new kernel. This gives the method of reproducing kernels for the reconstruction of visual surfaces more flexibility than the direct minimization methods (ala Grimson).

Although not presented here, the method can be easily extended to allow the information to be other than depth values (e.g derivatives, and/or integrals), and these forms of information may be intermingled. Also the method is easily modified to allow reconstruction in more dimension (e.g. density reconstruction in three dimensions for CAT scans).

We end with some of the drawbacks of the use of reproducing kernels to recover the splines. Some of the cons include: the assumptions necessary to allow the use of reproducing kernels (smothness of class and the norm); Reproducing kernels are currently know for only a small number of classes and associated norms); the possibly substantial amount of time to compute the parameters of the spline, especially if k is large. The complexity of some of the reproducing kernels themselves (e.g. two used by Atteia are infinite series) and others are very complicated formulas.

6 Example Classes, Kernels and Linear Systems.

In this section we exhibit 2 families of classes of surfaces, and their associated kernels. The semi-norms associated with these families of classes have as special cases the "quadratic variation" semi-norm used by [Grimson 81]. This semi-norm has the nice physical analogy of measuring the bending energy in a thin plate.

The first example is based on the work of [Duchon 76], and we include the parameterized class definition, parameterized form of the reproducing kernel, a particular example kernel, and the linear system (for calculation of the parameters of the spline σ_D with the particular kernel) for information points on a 4 by 4 unit spaced grid. The second example is based on the work of [Meinguet 79a,b, 80]. Here we only include a parameterized family of classes, the associated parameterized generating kernels, and one example of a particular kernel for information points on a 4–4 unit grid.

For the first family of classes we define $D^{-m}H^s$, for $s < 1$, to be the space of functions which have all derivatives of order m in H^s, where H^s is the Hilbert space of functions such that their tempered distributions v on $\Re^2$ having Fourier transform $\underline{v}$ that satisfy

$$\iint_{\Re^2} |\tau|^{2s} \; |\underline{v}\;(\tau)|^2 \, d\tau \quad < \infty$$

Let $F_1 = D^{-m}H^s$ be equipped with the semi-norm given by

$$\Psi_m(f) \equiv \sqrt{\sum_{i+j=m} \iint_{\Re^2} \left| D_x^i D_y^j \; f \right|^2}$$

Where $D^i_x f$ is $\partial^i f/\partial x^i$ and $D^i_y f$ is $\partial^i f/\partial y^i$. (Note that the null space of this seminorm, Π_{m-1}, is the space of bivariate polynomials of (total) degree $\leq$ m-1.) [Duchon 76] showed that if we require that $s < 1$ and $s > 1-m$ we have the reproducing kernels :

$$K_D((x,y),(s,t)) = ((x-s)^2 + (y-t)^2)^{(m+s-1)} \cdot \mathrm{Ln}\ [\sqrt{((x-s)^2 + (y-t)^2)}]$$

if 2(m+s-1) is an even positive integer

and

$$K_D((x,y),(s,t)) = ((x-s)^2 + (y-t)^2)^{(m+s-1)}$$

otherwise.

In particular, if we choose m = 2 (this makes the norm the surface variation norm used by Grimson) and $s = {}^1/_2$, we have the reproducing kernel $K_1(x,y).(s,t)) = ((x-s)^2 + (y-t)^2)^{3/2}$. Then we the coefficients of the spline σ_D are given by the solution of the linear system, A_D. An example of this system is given in Figure 1, where the the locations of the information samples are fixed at (i,j) i=1..4, j = 1..4, the associated depth values are $z_{i,j}$ and where for brevity we write x^{-} in place of $x^{(3/2)}$. Figures 2-7 contain example projections of surfaces from this space generated from either 16 or 100 information points.

Note the block toeplitz nature of the upper left corner of the matrix. We can take advantage of this very regular structure in the solution of A_D. Algorithms exists for the inversion of Block toeplitz systems in $O(k^{3/2})$, where k is the number of data points, (e.g. [Akaike 73] and [Watson 73]). Given the inverse of the block toeplitz portion of A_D, one can efficiently (in $O(k^2)$ time) precompute the basis splines, i.e. invert A_D. Then one can recover the coefficients for a particular set of data in $O(k^2)$ time by computing $A_D^{-1} \cdot [z_1,\ldots z_k]^T$. Recent work by Lee [Lee 85], gives an algorithm that soves a block toeplitz system on time $O(k^2 \log k)$, without inverting the matrix.

This toeplitz structure will occur only when the location of the information samples form a regular grid. However, the system is always symmetric. Unfortunately, it is always indefinite (in fact is always has $(m+1) \cdot m / 2$ negative eigenvalues). This may effect the choice of algorithm for the solution of the system. In fact the two block toeplitz inversion algorithms mentioned above maybe unstable for this system, because it is indefinite (They have not been shown unstable, but related algorithms for pointwise toeplitz systems are provably unstable for indefinite systems).

Another important property of a linear system is the condition number. This effects the accuracy to which *any computer algorithm* can solve the system. The larger the condition number, the less accurate a computer calculated solution to the system must be. The condition number of the matrix for the calculation of the spline coefficients depends on m, s, k. Furthermore for information samples on a regular grid, the condition number also depends on h, the separation between grid points. Initial results for the case s=0,m=2, and h appropriately chosen yield that the condition number of the system is approximately $3k^2$, where k is the number of information samples, and for the the case $s={}^1/_2$, m = 2 and h appropriately chosen, $\mathrm{Cond}(A_D) \approx 19k^2$. Both estimations from calculated values of the condition number for grid sized from 2^2 to 10^2 points, and h values taken from the range [.1 ,1.0].

$$
\begin{bmatrix}
0^+ & 1^+ & 4^+ & 9^+ & 1^+ & 2^+ & 5^+ & 10^+ & 4^+ & 5^+ & 9^+ & 13^+ & 9^+ & 10^+ & 13^+ & 18^+ & 1 & 1 & 1 \\
1^+ & 0^+ & 1^+ & 4^+ & 2^+ & 1^+ & 2^+ & 5^+ & 5^+ & 4^+ & 5^+ & 9^+ & 10^+ & 9^+ & 10^+ & 13^+ & 1 & 1 & 2 \\
4^+ & 1^+ & 0^+ & 1^+ & 5^+ & 2^+ & 1^+ & 2^+ & 9^+ & 5^+ & 4^+ & 5^+ & 13^+ & 10^+ & 9^+ & 10^+ & 1 & 1 & 3 \\
9^+ & 4^+ & 1^+ & 0^+ & 10^+ & 5^+ & 2^+ & 1^+ & 13^+ & 9^+ & 5^+ & 4^+ & 18^+ & 13^+ & 10^+ & 9^+ & 1 & 1 & 4 \\
1^+ & 2^+ & 5^+ & 10^+ & 0^+ & 1^+ & 4^+ & 9^+ & 1^+ & 2^+ & 5^+ & 10^+ & 4^+ & 5^+ & 9^+ & 13^+ & 1 & 2 & 1 \\
2^+ & 1^+ & 2^+ & 5^+ & 1^+ & 0^+ & 1^+ & 4^+ & 2^+ & 1^+ & 2^+ & 5^+ & 5^+ & 4^+ & 5^+ & 9^+ & 1 & 2 & 2 \\
5^+ & 2^+ & 1^+ & 2^+ & 4^+ & 1^+ & 0^+ & 1^+ & 5^+ & 2^+ & 1^+ & 2^+ & 9^+ & 5^+ & 4^+ & 5^+ & 1 & 2 & 3 \\
10^+ & 5^+ & 2^+ & 1^+ & 9^+ & 4^+ & 1^+ & 0^+ & 10^+ & 5^+ & 2^+ & 1^+ & 13^+ & 9^+ & 5^+ & 4^+ & 1 & 2 & 4 \\
4^+ & 5^+ & 9^+ & 13^+ & 1^+ & 2^+ & 5^+ & 10^+ & 0^+ & 1^+ & 4^+ & 9^+ & 1^+ & 2^+ & 5^+ & 10^+ & 1 & 3 & 1 \\
5^+ & 4^+ & 5^+ & 9^+ & 2^+ & 1^+ & 2^+ & 5^+ & 1^+ & 0^+ & 1^+ & 4^+ & 2^+ & 1^+ & 2^+ & 5^+ & 1 & 3 & 2 \\
9^+ & 5^+ & 4^+ & 5^+ & 5^+ & 2^+ & 1^+ & 2^+ & 4^+ & 1^+ & 0^+ & 1^+ & 5^+ & 2^+ & 1^+ & 2^+ & 1 & 3 & 3 \\
13^+ & 9^+ & 5^+ & 4^+ & 10^+ & 5^+ & 2^+ & 1^+ & 9^+ & 4^+ & 1^+ & 0^+ & 10^+ & 5^+ & 2^+ & 1^+ & 1 & 3 & 4 \\
9^+ & 10^+ & 13^+ & 18^+ & 4^+ & 5^+ & 9^+ & 13^+ & 1^+ & 2^+ & 5^+ & 10^+ & 0^+ & 1^+ & 4^+ & 9^+ & 1 & 4 & 1 \\
10^+ & 9^+ & 10^+ & 13^+ & 5^+ & 4^+ & 5^+ & 9^+ & 2^+ & 1^+ & 2^+ & 5^+ & 1^+ & 0^+ & 1^+ & 4^+ & 1 & 4 & 2 \\
13^+ & 10^+ & 9^+ & 10^+ & 9^+ & 5^+ & 4^+ & 5^+ & 5^+ & 2^+ & 1^+ & 2^+ & 4^+ & 1^+ & 0^+ & 1^+ & 1 & 4 & 3 \\
18^+ & 13^+ & 10^+ & 9^+ & 13^+ & 9^+ & 5^+ & 4^+ & 10^+ & 5^+ & 2^+ & 1^+ & 9^+ & 4^+ & 1^+ & 0^+ & 1 & 4 & 4 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 0 & 0 & 0 \\
1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \\ \alpha_9 \\ \alpha_{10} \\ \alpha_{11} \\ \alpha_{12} \\ \alpha_{13} \\ \alpha_{14} \\ \alpha_{15} \\ \alpha_{16} \\ \beta_1 \\ \beta_2 \\ \beta_3
\end{bmatrix}
=
\begin{bmatrix}
z_{11} \\ z_{12} \\ z_{13} \\ z_{14} \\ z_{21} \\ z_{22} \\ z_{23} \\ z_{24} \\ z_{31} \\ z_{32} \\ z_{33} \\ z_{34} \\ z_{41} \\ z_{42} \\ z_{43} \\ z_{44} \\ 0 \\ 0 \\ 0
\end{bmatrix}
$$

Figure 1 The matrix for the coefficients of the spline σ_D for information on a 4 by 4 grid. Where x^+ represents $x^{3/2}$, and z_{ij} is the information at grid point (i,j).

These comments on the indefiniteness of the system, and the rate of growth of the condition number reinforce the point that this method is currently only practical for *sparse* data.

A family of classes very closely related to that above was studied in [Meinguet 79a, 79b]. This family of classes (Beppo Levi spaces) is defined as follows:

$$X_m \equiv \left\{ f \in \mathcal{D} : D_{i_1} \ldots D_{i_m} f \in L^2 \text{ for } i_1, \ldots, i_m \in [1,2] \right\}$$

(Where $\mathcal{D}$ is the space of all Schwatrz distributions in $\Re^2$, and the partial derivatives are in the distributional sense). Intuitivly these spaces consist of those functions for which all partial derivatives up to (total) order m are square integrable. The mth Sobolev semi-norm is also the semi-norm for this family of spaces. Furthermore, the space X_m / Π_{m-1} is a Hilbert space, with the m^th^ Sobolev semi-norm as a true norm.. The reproducing kernel, E((x,y), (s,t)) associated with X_m is given by

$$E(\,(x,y),(s,t)\,) = c \cdot \left((x-s)^2 + (y-t)^2\right)^{(2(m-1))} \cdot \mathrm{Ln}\left(\sqrt{(x-s)^2 + (y-t)^2}\right)$$

where $c = \left(\ 2^{2m-1} \cdot \pi \cdot (m-1)! \cdot (m-1)!\ \right)^{-1}$.

Now if we define p_i to be the unique basis of Π_{m-1} such that $p_i(x_j,y_j) = \delta_{ij}$ for all $i,j \in J$ (the set of indices of the unisolvent subset of the information). Then we can express the reproducing kernel of X_m / Π_{m-1} as

$$K_M((x,y),(s,t)) = (-1)^m \cdot \Bigg(E((x,y),(s,t)) - \sum_{i \in J} p_i(x,y) \cdot E((x_i,y_i),(s,t))$$

$$- \sum_{i \in J} p_i(s,t) \cdot E((x,y),(s_i,t_i)) + \sum_{i \in J} \sum_{i \in J} p_i(x,y) \cdot p_j(s,t) \cdot E((x_i,y_i),(x_j,y_j)) \Bigg)$$

We now present a example in more detail. For concreteness we shall assume that the information samples fixed at (i,j) i=0..3, j = 0..3, and the associated depth values are $z_{i,j}$. We shall also chose m = 2, therefore our semi-norm is "quadratic variation". Since we must assume that that the information contains a unisolvent subset, we choose this subset to be at (0,0) (0,1) and (1,0) (and let the associates indices be 1,2,5 respectively). The associated $p_i(x,y)$ being given by $p_1(x,y) = 1.0 - (x+y)$, $p_2(x,y) = y$, and $p_3(x,y) = x$. Then the reproducing kernel for this class and distribution of information is given by

$$\begin{aligned} K_2(\,(x,y),(s,t)\,) = \{\ & E(\,(x,y),(s,t)\,) \\ & - (1-x-y) \cdot E((0,0),(s,t)) - (1-s-t) \cdot E((x,y),(0,0))\,) \\ & - y \cdot E((0,1),(s,t)) - t \cdot E((x,y),(0,1)) \\ & - x \cdot E((1,0),(s,t)) - s \cdot E((x,y),(1,0)) \\ & + (1-x-y) \cdot t \cdot E((0,0),(0,1)) + (1-x-y) \cdot s \cdot E((0,0),(1,0)) \\ & + y \cdot (1-t-s) \cdot E((0,1),(0,0)) + x \cdot (1-t-s) \cdot E((1,0),(0,0)) \\ & + y \cdot s \cdot E((0,1),(1,0)) + x \cdot t \cdot E((1,0),(0,1))\ \} \end{aligned}$$

Let us adopt the notation $A_M(m,k)$ to stand for the system of equations generated by (3.3) for the reproducing kernel for X_m / Π_{m-1}, and k = number of information samples, which are located on a regular grid (œk on a side.)

First we note that this system is always positive definite, and symmetric. Furthermore if we correctly order the variable we can insure that the Cholsky decomposition, written as $L_k L_k^T = A_M(m,k)$, has the property that L_k is the upper right hand corner of L_{k+1}, for all $i > 4$, see [Meinguet 79b] for a proof of this property. This implies that if we calculate the Cholesky decomposition for a larger square grid, we automatically have the decomposition for all smaller square grids. (In fact this can be used to define a recursive algorithm for the solution of the system, time cost $\approx O({}^1/_6 n^3)$). Given the decomposition, the actual coefficients for the spline can be recovered (in $O(n^2)$ time) using simple (and numerically stable) back-substitution.

The condition number of the systems $A_M(m,k)$ does not depend on the spacing of the points, and Initial experiments indicate that $\text{Cond}(A_M(2,k) \approx .95\ k^{2.5}$. This estimation based on calculation of the condition number for square grids of 16, 25, 36, 49, 64, 81, and 100 points.

7 Conclusions and Future Work.

This method seems particularly well suited to the problem if the user is interested in particularly smooth surfaces, and the initial information is sparse (in some sense the sparser the better). There is an added time and space savings if the location of the information samples can be fixed ahead of time, particularly in a regular structure.

Some of the work yet to be done includes:

Implementation of the current algorithm for more classes, and for non-fixed information samples;

Enhancements to the current implementation to allow information other than depth values;

Some justification (hopefully on psychological or psychophysical grounds) for the choice of classes;

Estimation of the condition number for more classes;

Full analysis of the numerical properties of the algorithm;

Through comparison of this method with existing methods;

An analysis of the approximation error of the algorithm.

References.

Akaike, Hirotugu, "Block Toeplitz Matrix Inversion", *SIAM J. Appl. Math.*, Vol. 24, No. 2, pp. 234-241, March 1973.

Atteia, M., "Functions "spline" et Noyaux reproduisants d'Aronszanj- Bergman", R.I.R.O., R-3, pp. 31-43, 1970.

Boult, Terrance, "Smoothness Assumptions in Human and Machine Vision, and Their Implications for Optimal Surface Interpolation.", In preperation.

Duchon, Jean, "Interpolation des Fonctions de Deux Variables Suivant le Principle de là Flexion des Plaques Minces", *Revue Française d'Automatique, Information, Recherche Opérationelle*, pp.5-12, 1976.

Duchon, Jean, "Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces". *Constructive Theory of Functions of Several Variables*, Oberwolfach 1976 (W. Schempp and K. Zeller ed.) pp. 85-100, Springer- Verlag, Berlin, Hiedelberg, 1977.

Grimson, W.E.L., *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, MA, (1981).

Kender, John, David Lee, and Terrance Boult, "Information-Based Complexity Applied to the 2 $\frac{1}{2}$ D Sketch", Columbia University Computer Science Technical Report CUCS149-85, (1985).

Lee, David, Personal Communication, 1985.

Meinguet, Jean, "Multivariatee Interpolation at Arbitrary Points Made Simple", *Journal of Applied Mathematics and Physics*, (ZPAM) 30, pp.292-304, 1979a.

Meinguet, Jean, "Basic Mathematical Aspects of Surface Spline Interpolation", ISNM 45: *Numerische Integration*, pp. 211-220, G. Hämmerilin ed., Basel: Birkhäuser Verlag 1979b.

Meinguet, Jean, "Surface Spline Interpolation : Basic Theory and Computational Aspects", Institut De Mathematique Pure et Appliquee, Universite Catholique de Louvain, Rapport n° 35, Oct. 1983.

Traub, J.T. and H. Wozniakowski, *A General Theory of Optimal Algorithms* , Academic Press NY (1980).

Traub, J.T., G. Wasilkowski, and H. Wozniakowski, *Information. Uncertainty and Complexity* , Addison Wesley, MA, (1983).

Watson, G.A., "An Algorithm for the Inversion of Block Matrices of Toeplitz Form", *Journal of the ACM*, Vol. 20, No. 3, pp. 409-415, July 1973.

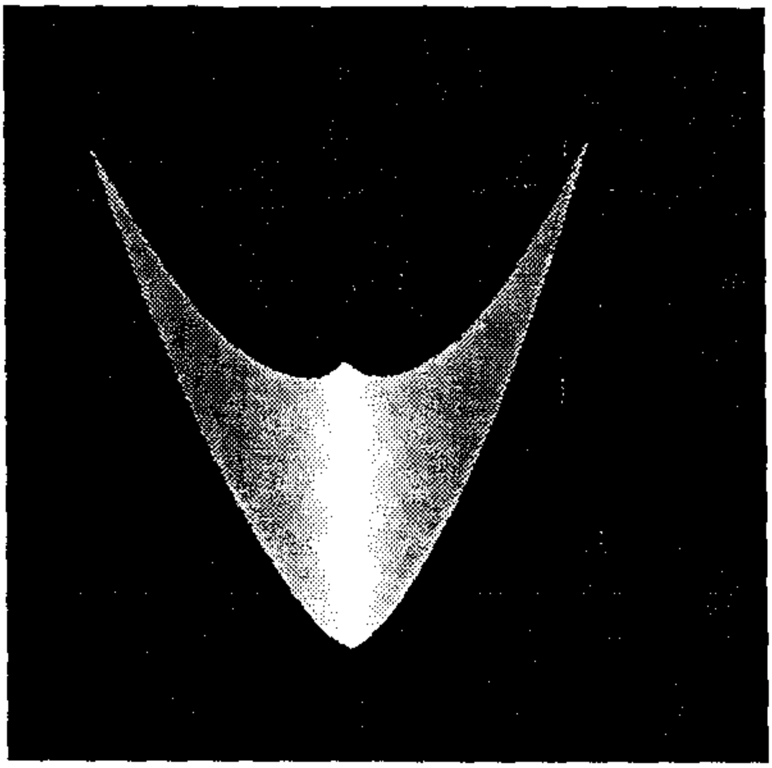

Figure 2: Interpolation of parabolic sheet. Reconstructed by evaluating surface at 48x48 points, and plotting tringular patches between the values. Input data taken at 16 points on a 4x4 square grid.

Figure 3: Data used for Figures 4 and 5. The input was on a 10x10 grid (lower left is (1,1)).

9	9	9	6	6	6	3	3	3	0
9	9	9	6	6	6	3	3	3	0
9	9	9	6	6	6	3	3	3	0
6	6	6	6	6	6	3	3	3	0
6	6	6	6	6	6	3	3	3	0
6	6	6	6	6	6	3	3	3	0
3	3	3	3	3	3	3	3	3	0
3	3	3	3	3	3	3	3	3	0
3	3	3	3	3	3	3	3	3	0
0	0	0	0	0	0	0	0	0	0

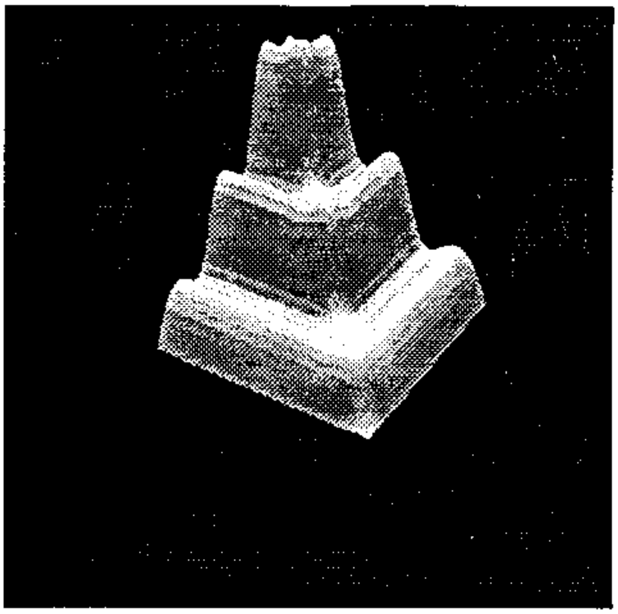

Figure 4: Interpolation of 1/4 weading cake frontal view. Reconstructed by evaluating surface at 48x48 points, and plotting tringular patches between the values. Input data given in Figure 3.

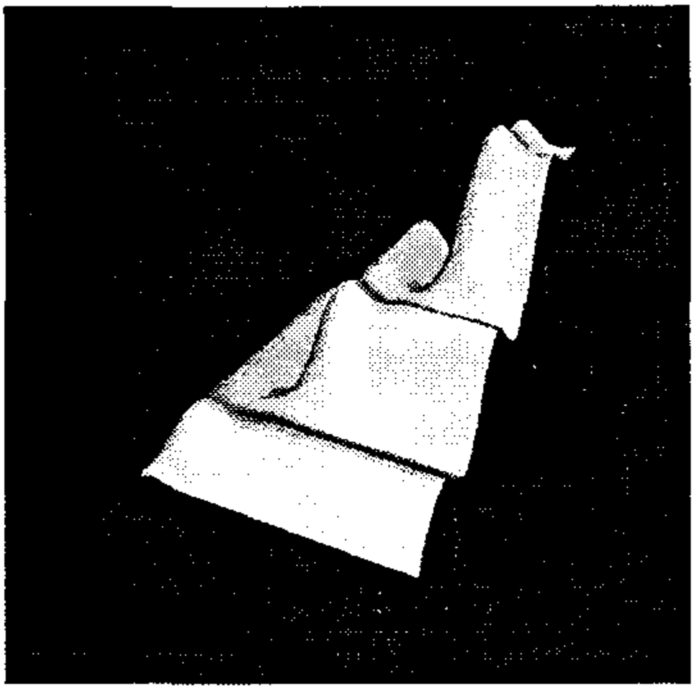

Figure 5: Another view of surface in Figure 4, highlighting the effect near edges.

Figure 6: Input data for Figure 7. The data is intended to suggest three planes floating in the air. Note that such a surface would not be in the class of surfaces that the algorithm, assumes, hence the papering over of the edges. There are 100 data points, laid out on a 10x10 square grid.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	5	5	5	5	5	5	0	0
0	0	5	5	5	5	5	5	0	0
0	0	5	5	10	10	5	5	0	0
0	0	5	5	10	10	5	5	0	0
0	0	5	5	5	5	5	5	0	0
0	0	5	5	5	5	5	5	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

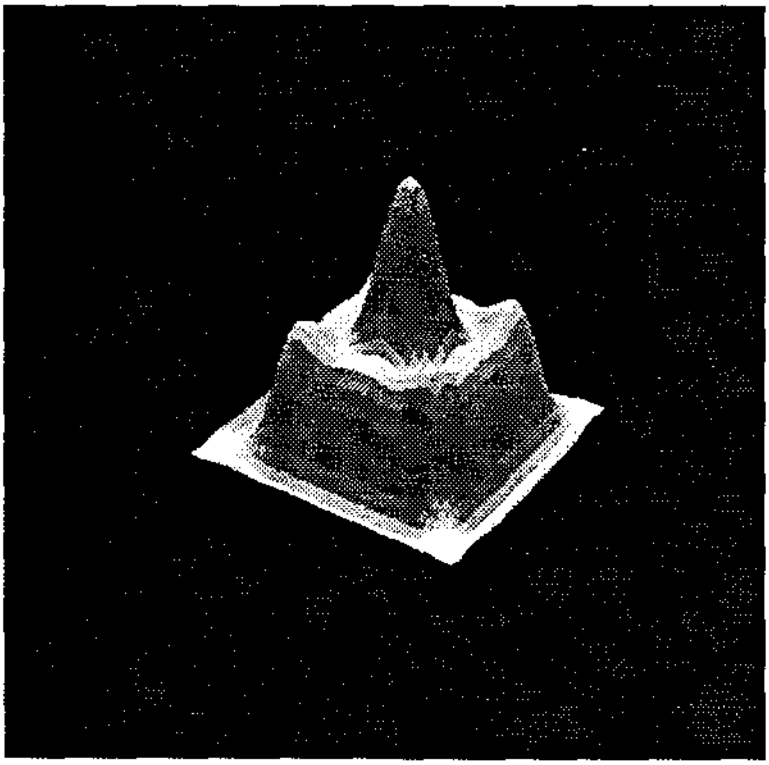

Figure 7: Interpolation of a “wedding cake” surface, Reconstructed by evaluating surface at 48x48 points, and plotting tringular patches between the values, Input data given in figure 6.