An Efficient Scattered Data Approximation Using Multilevel B-splines Based on Quasi-Interpolants

Byung-Gook Lee, Joon Jae Lee
Division of Computer and Information Engineering, Dongseo University, Busan 617-716, Korea
lbg,jjlee@dongseo.ac.kr

Jaechil Yoo
Department of Mathematics, Dongeui University, Busan 614-714, Korea
yoo@deu.ac.kr

Abstract

In this paper, we propose an efficient approximation algorithm using multilevel B-splines based on quasi-interpolants. Multilevel technique uses a coarse to fine hierarchy to generate a sequence of bicubic B-spline functions whose sum approaches the desired interpolation function. To compute a set of control points, quasi-interpolants gives a procedure for deriving local spline approximation methods where a B-spline coefficient only depends on data points taken from the neighborhood of the support corresponding the B-spline. Experimental results show that the smooth surface reconstruction with high accuracy can be obtained from a selected set of scattered or dense irregular samples.

1 Introduction

Recently, there are many interesting approaches for reconstructing smooth 3D surface from discrete uniform data points or scattered data points. The problem of reconstructing smooth surfaces arises in many fields of science and engineering, and the data sources include measured values such as laser range scanning.

The problem of recovering a surface from a set of data is simple in concept but tricky when we get into the detail. Since the real world is made up of continuous surfaces, not discrete points, we want to create a continuous surface from the unorganized data points. The ultimate goal of this paper is to find a surface reconstruction method as getting a smooth and high fidelity of 3D surface from a large numbers of scattered data points. In particular, the description should be sufficiently completed to reconstruct the 3D surface within a certain tolerance error, given their relative locations and expected noise.

There exist many techniques for surface approximation to improve the approximate continuity and smoothness in handling a large number of data. Tensor product of B-splines surfaces is widely used to approximate rather than to work with other types of approximation because of the advantages inherent in working with tensor products. Tensor product guarantees internal continuity if the knot vectors are set properly.

Multilevel idea has been adopted to reduce the approximation error. Therefore this paper is based on the multilevel B-splines approximation techniques presented in 1997, the publication of Lee, Wolberg and Shin[7]. They named the schemes *multilevel B-splines*. In the previous work, Forsey and Bartels[4] developed a surface fitting method which is adaptive on hierarchical spline functions. However, this method cannot deal with scattered data. Lee presented a multilevel B-spline algorithm to fit a uniform bicubic B-spline surface to scatterd data where multilevel or hierarchy is used to reduce the approximation errors. The method does not guarantee a reasonable global approximation at initial level even though it has an advantage of local processing.

The splines approximation technique used in this paper is quasi-interpolants, first developed by de Boor and Fix[2]. The quasi-interpolants operators were later generalized by Lyche and Schumaker[5], and it was their version that used in the alternative surface approximation technique. A quasi-interpolants operator approximates a curve by calculating coefficients that are used to weight samplings of the curve to be approximated. Lyche and Schumaker quasi-interpolants operator uses coefficients that are inexpensive to calculate and samplings that are relatively expensive to calculate. It turns out to produce splines approximation with the required accuracy. According to the trend in recent algorithms, the hierarchical, multiresolution technique has been used for scattered data and irregular samples.

Proceedings of the 5th Int'l Conf. on 3-D Digital Imaging and Modeling (3DIM 2005)
1550-6185/05 $20.00 © 2005 IEEE

A new algorithm using quasi-interpolants is introduced to implement the multilevel B-spline approximation and apply to both scattered data and dense data. The proposed method is fast for data interpolation and approximation, maintaining the accuracy. This algorithm achieves a C^2-continuous interpolation function from arbitrary scattered data and can process huge numbers of data with numerically stable. Quasi-interpolants give a procedure for deriving local spline approximation methods where a B-spline coefficient only depends on data points taken from the neighborhood of the support corresponding B-spline.

In range data analysis, surface reconstruction from large number of data sets is very challenging, especially if the data present undesired holes and missing. This is usually the case when the data come from laser scanner for 3D acquisitions or if they represent damaged objects to be restored. In this work, a local approach to surface reconstruction from the range data based on this algorithm is presented that fills the holes and interpolate missing data points very smoothly with approximation. The local nature of the algorithm allows for fast computational time of large amounts of data, since the computation is limited to local region.

More explanation about multilevel B-splines approximation which can be used to generate an interpolation function through scattered data points is described in section 2. Section 3 gives the explanation on how to reconstruct the quasi-interpolants. Then, section 4 shows the experimental results for numerical examples and finally, conclusions are given in section 5.

2 Multilevel B-spline Approximation

The methods explored in this paper take a set of scattered data as input and produce tensor product B-spline surfaces as output. The algorithms run in a multiresolutional setting over uniform partitions such that the final surface f is composed of a sequence of surfaces at dyadic scales,

$$f = f_0 + f_1 + \ldots + f_k,$$

where $f_i \in S_i, i = 0, 1, \ldots, k$, and $S_0, S_1, \ldots, S_k$ is a nested sequence of subspaces of S_k,

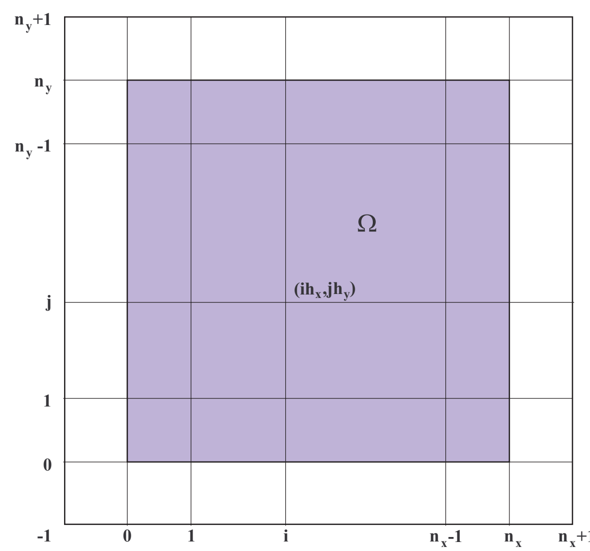
$$S_0 \subset S_1 \subset \ldots \subset S_k.$$

The basic algorithms used for the results presented in this paper were published in 1997 by Lee, Wolberg and Shin. They called the schemes *Multilevel B-splines*. Our interest is mainly scattered data interpolation and approximation, which is also the main focus in [7].

Given a set of scattered points $P = \{P_i\}_{i=1}^n$, $P_i = (x_i, y_i, z_i) \in R^3$ and let $\Omega = \{(x,y)|0 \le x < m_x, 0 \le y < m_y\}$ be a rectangular domain in the xy-plane such that (x_i, y_i) is a point in Ω. Let Φ be a control lattice overlaid on a domain Ω. The control lattice Φ is an uniform tensor product grids over Ω.

To approximate scattered data points P, we formulate initial approximation function f as a uniform bicubic B-spline function, which is defined by a control lattice Φ. Let the initial number of control points on the lattice as $n_x = m_x/h_x$ in x-axis, and $n_y = m_y/h_y$ in y-axis. The knot intervals are uniform interval defined as h_x in x-axis and h_y in y-axis. So, for uniform cubic B-spline case, degree $d = 3$ and the set of knot vectors are defined as below:

$$\tau_x = \{-dh_x, \ldots, 0, h_x, \ldots, n_x h_x, \ldots, (n_x + d)h_x\}$$
$$\tau_y = \{-dh_y, \ldots, 0, h_y, \ldots, n_y h_y, \ldots, (n_y + d)h_y\}.$$



Ω

Let c_{ij} be the value of the ij-th control point on lattice Φ, located at position (ih_x, jh_y) of the grid defined by Φ, for $i = -1, 0, 1, \ldots, n_x + 1$ and $j = -1, 0, 1, \ldots, n_y + 1$. The approximation function f defined in terms of these control points at position $(x, y) \in \Omega$ is given as

$$f(x,y) = \sum_{i=1}^{n_x+1} \sum_{j=1}^{n_y+1} c_{ij} B_{i,d}(x) B_{j,d}(y) \tag{1}$$

where $B_{i,d}$ and $B_{j,d}$ are uniform cubic B-spline basis functions, $d = 3$ and knot vector for cubic B-spline basis are below:

$$\{(i-2)h_x, (i-1)h_x, ih_x, (i+1)h_x, (i+2)h_x\}$$
$$\{(j-2)h_y, (j-1)h_y, jh_y, (j+1)h_y, (j+2)h_y\}.$$

B-spline approximation(BA) algorithm generates a tradeoff that exists between the shape smoothness and accuracy of the approximation function. To overcome this tradeoff, multilevel B-splines approximation(MBA) algorithm is introduced [7]. The algorithm makes use of a hierarchy of control lattices to generate a sequence of function f_i and the final approximation function f is defined as the sum of functions f_i,

$$f = \sum_{i=1}^{k} f_i. \tag{2}$$

To optimize this process, B-spline refinement is used to reduce the sum of these functions into one equivalent B-spline function. The MBA algorithm serves result as smooth initial approximation f_0 to $\Delta^0 P = P$ defined on the coarsest control lattice $\Phi_0 = \Phi$, by applying the BA algorithm. To continue to the finer levels, below explanation are quoted from [7]: The first approximation possibly leaves large discrepancies at the data points in P. In particular, f_0 leaves a deviation

$$\Delta^1 z_i = z_i - f_0(x_i, y_i) \text{ for } i = 0, \ldots, n. \tag{3}$$

The next finer control lattice Φ_1 is then used to obtain function f_1 that approximates the difference $\Delta^1 P = \{(x_i, y_i, \Delta^1 z_i)\}$.

Then, the sum of $f_0 + f_1$ yields a smaller deviation (3) for each (x_i, y_i) in Ω.

$$\Delta^2 z_i = z_i - f_0(x_i, y_i) - f_1(x_i, y_i) \text{ for } i = 0, \ldots, n.$$

In general, for each level k in the hierarchy, the point set $\Delta^k P = \{(x_i, y_i, \Delta^k z_i)\}$ is approximated by a function f_k defined over the control lattices Φ_k, where

$$\Delta^k z_i = z_i - \sum_{l=0}^{k-1} f_l(x_i, y_i) = \Delta^{k-1} z_i - f_{k-1}(x_i, y_i)$$

and $\Delta^0 z_i = z_i$. This process starts from the coarsest lattice Φ_0 and continue incrementally to the finest lattice Φ_k with the set of knot vectors are defined as below:

$$\tau_x^k = \{-d\frac{h_x}{2^k}, \ldots, 0, \frac{h_x}{2^k}, \ldots, 2^k n_x \frac{h_x}{2^k}, \ldots, (2^k n_x + d)\frac{h_x}{2^k}\}$$
$$\tau_y^k = \{-d\frac{h_y}{2^k}, \ldots, 0, \frac{h_y}{2^k}, \ldots, 2^k n_y \frac{h_y}{2^k}, \ldots, (2^k n_y + d)\frac{h_y}{2^k}\}.$$

The final approximation function f is defined as the sum of the functions (2). They are many methods for refining a control lattice into another so that they generate the same B-spline functions. In this paper, B-spline refinement of an $(n_x+3) \times (n_y+3)$ control lattice $\Phi_0 = \Phi$ is always refined to a $(2^k n_x + 3) \times (2^k n_y + 3)$ control lattice Φ_k whose the control point spacing is half.

3 Quasi-interpolants

Many applications of splines make use of some approximation method to produce a spline function from given discrete data. Popular methods include interpolation and least squares approximation. However, both of these methods require solution of a linear system of equations with as many unknowns as the dimension of the spline space, and are therefore not suitable for real-time processing of large streams of data. For this purpose local methods, which determine spline coefficients by using only local information, are more suitable. To ensure good approximation properties it is important that the methods reproduce polynomials and preferably the functions in the given spline space. A method based on derivative information was constructed in [2], while a more general class was studied in [5]. In order to reproduce the spline space, the local information of the methods in [5] was restricted to lie in one knot interval. In this paper we remove this restriction. We then discuss some specific approximation methods for quadratic and cubic splines.

We use B-splines as a basis for splines and denote the i^{th} B-spline of degree d with knots τ by $B_{i,d} = B_{i,d,\tau}$, and the linear space spanned by these B-splines by $S_{d,\tau}$.

Given a function f, the basic problem of spline approximation is to determine B-spline coefficients $(c_i)_{i=1}^n$ such that

$$Pf = \sum_{i=1}^{n} c_i B_{i,d}$$

is a reasonable approximation to f. The basic challenge is therefore to devise a procedure for determining the B-spline coefficients. We assume that f is defined on an interval $[a, b]$, and that we have selected a space of splines $S_{d,\tau}$ defined on $[a, b]$ (i.e., so that $\tau = (t_j)_{j=1}^{n+d+1}$ is nondecreasing with $t_{d+1} = a$ and $t_{n+1} = b$). We fix k and propose the following procedure for determining c_k:

(i) Choose a local interval $I = (t_\mu, t_\nu)$ with the property that I intersects the (interior of the) support of $B_{k,d}$:

$$I \cap (t_k, t_{k+d+1}) \neq \emptyset.$$

Denote the restriction of the space $S_{d,\tau}$ to the interval I by $S_{d,\tau,I}$, i.e.,

$$S_{d,\tau,I} = \text{span}\{B_{\mu-d,d}, \ldots, B_{\nu-1,d}\}.$$

(ii) Choose some local approximation method P_I with the property that

$$P_I g = g, \text{ for all } g \in S_{d,\tau,I}. \tag{4}$$

Proceedings of the 5th Int'l Conf. on 3-D Digital Imaging and Modeling (3DIM 2005)
1550-6185/05 $20.00 © 2005 IEEE

(iii) Let f_I denote the restriction of f to the interval I. Then there exist B-spline coefficients $(b_i)_{i=\mu-d}^{\nu-1}$ such that $P_I f_I = \sum_{i=\mu-d}^{\nu-1} b_i B_{i,d}$. Note that $\mu - d \le k \le \nu - 1$ since supp$B_{k,d}$ intersects I.

(iv) Set $c_k = b_k$.

When determining c_k, this procedure gives us the freedom to restrict our attention to a local subinterval $I = [t_\mu, t_\nu]$ of our choice. By doing this we may reduce the complexity of the problem. Secondly, we have the freedom to choose the local approximation method P_I. Typical choices will be interpolation, least squares approximation, or a smoothing spline. As we shall see in Lemma 1, the local condition (4) ensures that if f is a spline in $S_{d,\tau}$, it will be reproduced by Pf. In certain situations, other conditions may be more natural, but we will not pursue this any further here.

We first ascertain that the local reproduction condition leads to global reproduction of $S_{d,\tau}$.

LEMMA 1 *The spline approximation Pf determined by steps (i)–(iv) above has the property that $Pf = f$ for all f in the spline space $S_{d,\tau}$.*

To emphasize the dependence on f, the coefficient c_k is often written $c_k = \lambda_k f$, with λ_k some linear functional. The following lemma gives an explicit formula for the coefficient $\lambda_k f$ in the case where it is a combination of given linear functionals $\lambda_{k,1}, \ldots, \lambda_{k,\nu-\mu+d}$.

LEMMA 2 *Suppose that the coefficient c_k of Pf is chosen as*

$$c_k = \frac{\det\big(\lambda B_{\mu-d}, \ldots, \lambda B_{k-1}, \lambda f, \lambda B_{k+1}, \ldots, \lambda B_{\nu-1}\big)}{\det\big(\lambda B_{\mu-d}, \ldots, \lambda b_{\nu-1}\big)}, \quad (5)$$

where λB_j denotes the column vector

$$\lambda B_j = (\lambda_{k,1} B_{j,d}, \ldots, \lambda_{k,\nu-\mu+d} B_{j,d})^T$$

and $\lambda_{k,1}, \ldots, \lambda_{k,\nu-\mu+d}$ are linear functionals defined on $S_{d,\tau}$ such that the denominator in (5) is nonzero. Then $Pf = f$ for all f in $S_{d,\tau}$.

A general class of approximation methods are obtained by letting P_I be given as point functionals of the form

$$\lambda_{k,j} f = f(x_{k,j}) \text{ for } j = 1, \ldots, m_k,$$

where $m_k = \nu - \mu + d$ and $x_{k,1}, \ldots, x_{k,m_k}$ are given points. With this choice, it is well known (see page 200 of [1]) that if

$$B_{\mu-d-1+j,d}(x_{k,j}) > 0 \text{ for } j = 1, \ldots, m_k,$$

then the denominator in (5) is nonzero and Lemma 2 can be applied. Expanding the numerator in (5), we obtain c_k in the form

$$c_k = \lambda_k f = \sum_{j=1}^{m_k} w_{k,j} f(x_{k,j}), \quad (6)$$

for some vector $w_k = (w_{k,j})$. Equivalently, we can find w_k by solving the linear system

$$\delta_{i,k} = \lambda_k(B_{i,d}) = \sum_{j=1}^{m_k} w_{k,j} B_{i,d}(x_{k,j}), \quad (7)$$

for $i = \mu - d, \ldots, \nu - 1$ where $\delta_{i,k} = 1$ if $i = k$ and zero otherwise, as usual. In practice one would usually determine c_k numerically, either from (4), or (5), except in special cases where the formulas are particularly simple.

Quasi-interpolants of this kind were studied in [5]. However, there the data points $\{x_{k,j}\}_{j=1}^{m_k}$ are restricted to all lie in one subinterval $[t_l, t_{l+1}]$ of $[t_k, t_{k+d+1}]$.

There are standard ways to obtain error estimates for the kind of approximation methods developed here. Let us denote the total approximation by Pf, and suppose we have found a constant C (that may depend on the knots, but not on f) such that

$$\|Pf\| \le C\|f\|. \quad (8)$$

Here $\|f\|$ denotes the uniform norm on the interval $[a, b]$,

$$\|f\| = \max_{x\in[a,b]} \big|f(x)\big|.$$

From (8) it follows by a standard argument that

$$\|f - Pf\| \le (1 + C)\mathrm{dist}(f, S_{d,\tau}),$$

where $\mathrm{dist}(f, S_{d,\tau})$ denotes the quantity

$$\mathrm{dist}(f, S_{d,\tau}) = \inf_{g\in S_{d,\tau}} \|f - g\|.$$

We consider some examples in the case where the knots and the degree of the spline are given.

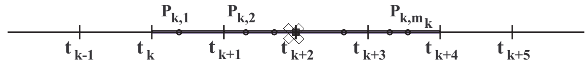
EXAMPLE 1 *In the cubic spline case ($d = 3$). To determine coefficient c_k, we choose the interval $I = [t_k, t_{k+4}]$ which means that the local spline space has dimension 7,*

$$S_{d,\tau,I} = \mathrm{span}\{B_{k-3,d}, B_{k-2,d}, \ldots, B_{k+3,d}\}.$$

Here, the data points $\{P_{k,i}\}_{i=1}^{m_k}, P_{k,i} = (x_{k,i}, y_{k,i}) \in R^2$ are restricted to lie in the interval $I = [t_k, t_{k+4}]$

Coefficient matrix :

$$\begin{bmatrix} B_{k-3,3}(x_{k,1}) & B_{k-3,3}(x_{k,2}) & \ldots & B_{k-3,3}(x_{k,m_k}) \\ B_{k-2,3}(x_{k,1}) & B_{k-2,3}(x_{k,2}) & \ldots & B_{k-2,3}(x_{k,m_k}) \\ \vdots & \vdots & \ddots & \vdots \\ B_{k+3,3}(x_{k,1}) & B_{k+3,3}(x_{k,2}) & \ldots & B_{k+3,3}(x_{k,m_k}) \end{bmatrix}$$



$$I = [t_\mu, t_\nu] = [t_k, t_{k+4}]$$

To compute the coefficient of the control points c_k, the Moore-Penrose inverse for minimum norm least-squares solution, is used. After the computation of least-squares solution, we get a set of $w_{k,j}$. We have to choose the middle one from the set of c_k for the control point value.

If the data is uniform, we don't need to compute the weight for each c_k, resulting in fast calculation time. This is the examples of weighted $w_{k,j}$: For $m_k = 5$,

$$[\frac{156}{1459}, -\frac{166}{1459}, \frac{2520}{1459}, -\frac{166}{1459}, \frac{156}{1459},]$$

and for $m_k = 9$,

$$[-\frac{134}{3299}, \frac{1072}{3299}, -\frac{15997}{19794}, \frac{2884}{9897}, \frac{2884}{9897}, -\frac{15997}{19794}, \frac{1072}{3299}, -\frac{134}{3299}].$$

The tensor product of the two spline spaces is defined to be a family of all functions of the form

$$(Pf)(x,y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} c_{ij} B_{i,d}(x) B_{j,d}(y)$$

where $B_{i,d}$ and $B_{j,d}$ are the B-splines on $\tau_x = (t_j)_{j=1}^{n_x+d+1}$ and $\tau_y = (s_j)_{j=1}^{n_y+d+1}$ respectively.

EXAMPLE 2 *In the tensor product cubic spline case ($d = 3$). To determine coefficient c_{ij}, we choose the interval $I = [t_\mu, t_\nu] \times [s_\mu, s_\nu] = [t_i, t_{i+4}] \times [s_j, s_{j+4}]$ which means that the local spline space has dimension 49,*

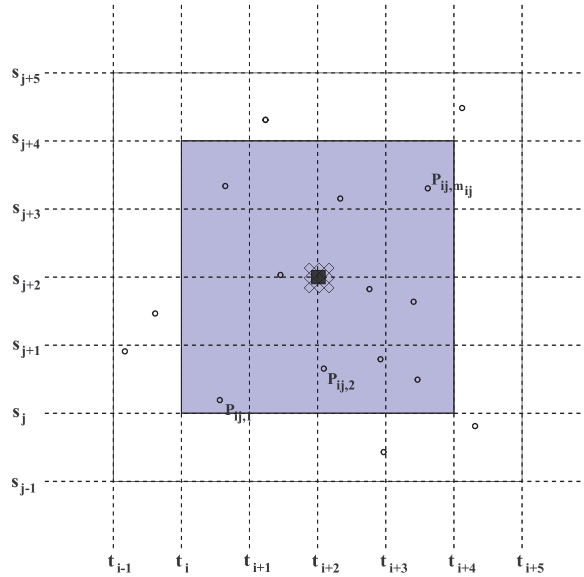
$$S_{d,\tau,I} = \text{span}\{B_{i-3,d}B_{j-3,d}, \ldots, B_{i+3,d}B_{j+3,d}\}.$$

Here, the data points $\{P_{ij,k}\}_{k=1}^{m_{ij}}, P_{ij,k} = (x_{ij,k}, y_{ij,k}, z_{ij,k}) \in R^3$ are restricted to lie in the interval $I = [t_i, t_{i+4}] \times [s_j, s_{j+4}]$

Coefficient matrix :

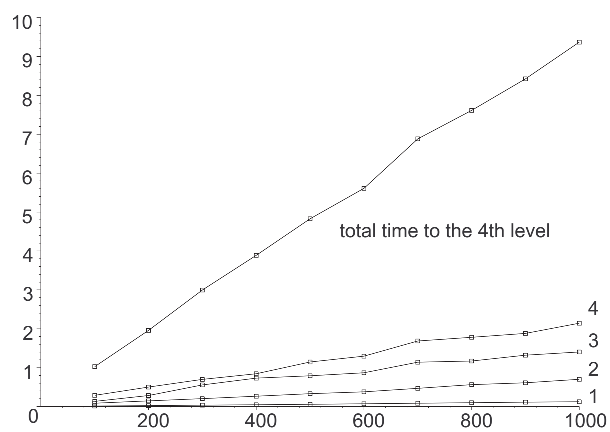
$$\begin{bmatrix} B_{i-3,3}(x_{ij,1})B_{j-3,3}(y_{ij,1}) & \cdots & B_{i-3,3}(x_{ij,m_{ij}})B_{j-3,3}(y_{ij,m_{ij}}) \\ B_{i-3,3}(x_{ij,1})B_{j-2,3}(y_{ij,1}) & \cdots & B_{i-3,3}(x_{ij,m_{ij}})B_{j-2,3}(y_{ij,m_{ij}}) \\ \vdots & \ddots & \vdots \\ B_{i+3,3}(x_{ij,1})B_{j+3,3}(y_{ij,1}) & \cdots & B_{i+3,3}(x_{ij,m_{ij}})B_{j+3,3}(y_{ij,m_{ij}}) \end{bmatrix}$$

Let n be the number of data points in P and let $(n_x + 3) \times (n_y + 3)$ be the size of the initial control lattice Φ_0. The number of control points in lattice Φ_k is $(2^k n_x + 3) \times (2^k n_y + 3)$. Hence, the time complexity of the MBA algorithm is $\mathrm{O}(n_x n_y) + \mathrm{O}(4 n_x n_y) + \ldots + \mathrm{O}(2^{k+1} n_x n_y) =$



$$I = [t_\mu, t_\nu] \times [s_\mu, s_\nu] = [t_i, t_{i+4}] \times [s_j, s_{j+4}]$$

$\mathrm{O}((2^{k+2} - 1) n_x n_y)$. To demonstrate the time efficiency of the proposed method, we calculate time as the sample points increase from 100 to 1000 by 100 steps where the number of control points at initial level is 7×7 and four levels are performed. Fig. 4 plots the relative time value to 100 sample points according to the sample size. From this result, we can see it linearly increases with sample size. The space complexity is $\mathrm{O}(n + (2^{k+2} - 1) n_x n_y)$ because we have to store all the control lattices in the hierarchy. But if an adaptive control lattice hierarchy is used, a control lattice can be reduced by a simple set of necessary control points.



4 Experimental Results

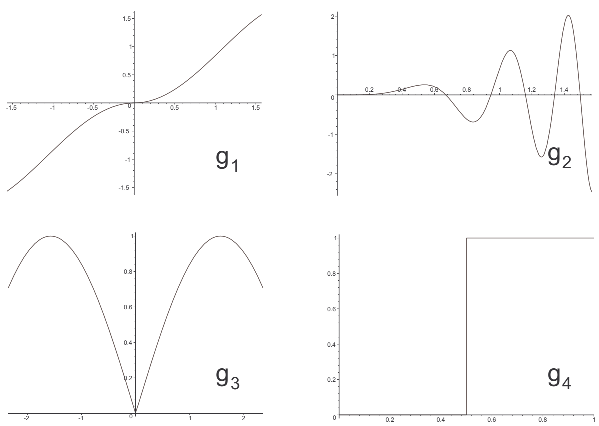
To demonstrate the accuracy of reconstruction by the proposed algorithm, we performed experiments with the various functions, images and real range data sets. First, our proposed quasi-interpolants is compared with the global method for one dimensional functions. Given a test function $g_k(x)$, we first sampled data points from it and applied to the algorithm to obtain an approximation function f. The difference between g_k and f is then measured by computing the normalized RMS(root mean square) error which is divided the RMS error by the difference of maximum and minimum values of g_k between the function values on a dense grid. That is,

$$RMS = \sqrt{\frac{\sum_{i=0}^{N}(g_k(x_i) - f(x_i))^2}{(N+1)}}$$

where $x_i = i/N$, and $N = 50$.

The test functions are

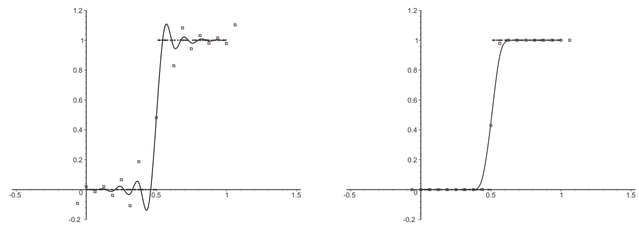
$$\begin{aligned} g_1(x) &= |x|sin(x) \\ g_2(x) &= x^2 sin(7x^2) \\ g_3(x) &= |sin(x)| \\ g_4(x) &= \begin{cases} 0 & \text{if } x \leq 0.5, \\ 1 & \text{if } x > 0.5. \end{cases} \end{aligned}$$



For each test function as shown in Fig. 5, we used 100 data randomly sampled. According to Table 1, the quasi-interpolants generates a reasonable approximation as good as global approximation for smooth functional curve regardless of type of the test functions where 5, 9 and 17 are the number of the control points. We note that the shape of the quasi-interpolants looks better than one of the global approximation for the step function, while the RMS error of the quasi-interpolants is greater than that of the global approximation as shown in Fig. 6.

Table 1. Normalized RMS errors.

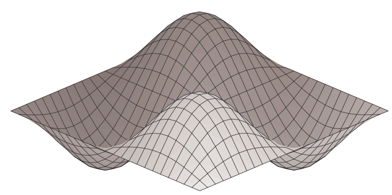
global	g_1	g_2	g_3	g_4
5	.7442964	.9196416	.0496652	.1640436
9	.7445050	.8493500	.0302326	.1251580
17	.7445548	.7991659	.0257063	.0880861
quasi	g_1	g_2	g_3	g_4
5	.7442964	.9196416	.0496652	.1640436
9	.7446013	.8462046	.0335381	.1388721
17	.7445674	.7988042	.0261447	.1079658



A) Global approximation B) Quasi-interpolants

To demonstrate the performance of multilevel B-spline approximation using quasi-interpolants, the following test function is used as shown in Fig 7.

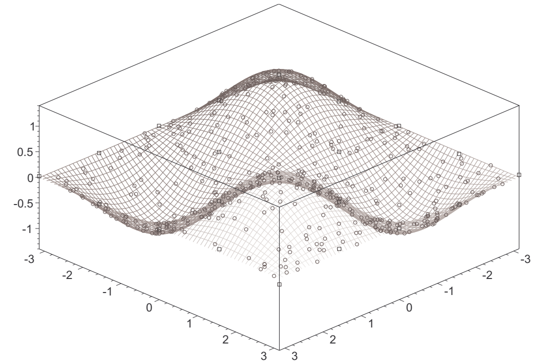
$$g(x, y) = sin(x) + sin(y).$$



We used three data sets of M100, M500 and R500, where M100 and M500 are small and large data sets, which consist of 100 and 500 points, respectively. We uniformly sampled 7×7 and 15×15 data points, respectively, while the others was randomly sampled. And R500 points were totally randomly sampled.

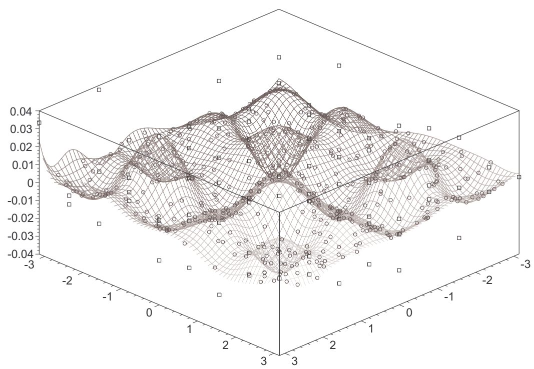
Fig. 8 and 9 shows approximation surface and error surface of g at initial level, respectively, where the circles represent the sampled data. The second level approximation result is obtained from sum of Fig. 8 and 9 as shown in Fig. 10. Table 2 represents that the proposed method recon-

Proceedings of the 5th Int'l Conf. on 3-D Digital Imaging and Modeling (3DIM 2005)
1550-6185/05 $20.00 © 2005 IEEE

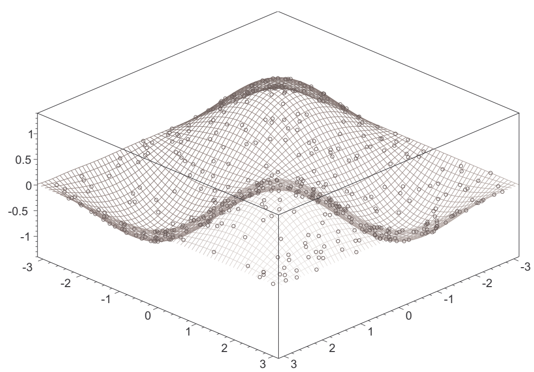


g

structs test functions very accurately within a few level. We started from the number of control points of 7x7 at initial level to three levels.



$\Delta^1 z = z_i - f_0(x_i, y_i)$



$f_0 + f_1$

The experiment is also performed for large number of uniform data such as flower image with size of 512 by 512. PSNR($20 \log(255/RMS)$)is used as a error metric for image compression. High quality reconstructed image of 31.933dB is obtained with $18.24 : 1$ of compression ratio at the level 5 in Fig. 11.

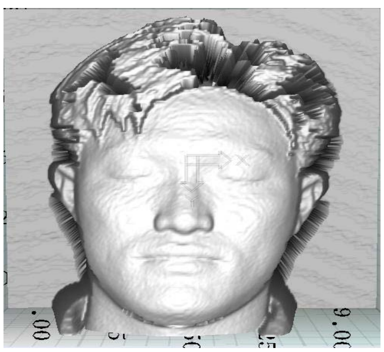
Table 2. Normalized RMS errors between test functions and their approximations.

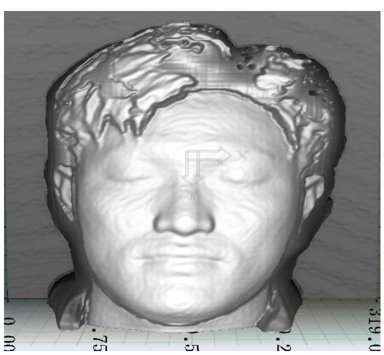
level	M100	M500	R500
1	.0049298	.0040621	.0041634
2	.0020481	.0001827	.0004930
3	.0017656	.0001323	.0004057

A real range data acquired from 3D scanner is used to prove the efficiency of the proposed algorithm. Fig. 12 shows a head data with many holes in hair region due to low reflection of laser where the size of data is 320×320 and the pixel intensities depict depth values. The undesired hole or missing points are usually case when the data come from laser scanner 3D acquisitions or if they represent damaged objects to be restored. An attractive field of research focuses on situations in which these holes are too geometrically and topologically complex to fill and entirely scattered data set exist. As shown in Fig. 13, the method generates good approximation and smooth surface, filling missing data points by interpolation property where the algorithm generated five level with initial 8×8 control points. Therefore it can be applied to many applications such as range data analysis by generating a smooth surface with fast convergence speed and linear calculation time.

5 Conclusion

This paper focuses on multilevel B-spline approximation based on quasi-interpolants for scattered data approximation and interpolation. The algorithm is fast and generates a C^2-continuous surface through a set of unevenly spaced points. Experimental results reveal that smooth 3D object reconstruction is possible from scattered data and irregular samples. Multilevel B-spline approximation was presented to circumvent the tradeoff which exists between the shape smoothness and approximation accuracy of the function, depending on the control lattice density. Then, quasi-interpolants was introduced to reduce the time complexity and memory usage in the system development especially when we deal with large number of range data. It is effectively gains in large performance. The quasi-interpolants is a special case of more general constructions and performs better approximation to reduces error results.





This work was supported by grant No.R01-2004-000-10851-0, R-05-2004-000-10968-0 from Ministry of Science & Technology.

References

[1] de Boor, C., A Practical Guide to Splines, Springer Verlag, New York, 1978.

[2] de Boor, C. and G.J.Fix, Spline approximation by quisiinterpolants, J. Approx. Theory, 8, (1973) 19-45.

[3] Byung-Gook Lee, Lyche, T. and Knut Mørken, Some Examples of Quasi-Interpolants Constructed from Local Spline Projectors, Mathematical Methods for Curves and Surfaces, Oslo 2000, Lyche, T. and Schumaker L.L (eds), Vanderblit Press, Nashville, pp. 243-252, 2001.

[4] David Forsey and Richard Bartels, Surface fitting with hierarchical splines, ACM Trans. on Graphics, 14, (1995) 134-161.

[5] Lyche, T. and L.L.Schumaker, Local spline approximation methods, J. Approx. Theory 25 (1979), 266-279.

[6] Øyvind Hjelle, Approximation of Scattered Data with Multilevel B-splines, Technical Report STF42 A01011, SINTEF 2001.

[7] S. Y. Lee, G. Wolberg, and S. Y. Shin. Scattered Data Interpolation with Multilevel B-Splines. IEEE Transactions on Visualization and Computer Graphics, 3(3), (1997) 229-244.