Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100128

B-Spline Surface Fitting on Scattered Points

Khang Jie Liew *, *Ahmad Ramli and Ahmad Abd. Majid*

School Mathematical Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia

Received: 27 Jun. 2015, Revised: 25 Aug. 2015, Accepted: 26 Aug. 2015
Published online: 1 Jan. 2016

Abstract: This paper looks into the effectiveness of B-spline approximation algorithm in approximating the bicubic B-spline surface from the set of scattered data points which are taken from the scanned 3D object in the form of point sets. Using the B-spline approximation algorithm, the unknown B-spline control points are determined, followed by the reconstruction of the bicubic B-spline surface. Using a set of neighbourhood of data points, a B-spline surface patch may be constructed, which can be pieced together to form the final surface. Modification of the B-spline approximation algorithm is carried out before the reconstruction in order to fit the scattered data points closely. Here, the density of the data points is scaled down due to the sparseness of the points that may affect the smoothness. The sample of scattered data points is chosen from a specific region in the point set model by using *k*-nearest neighbour search method. Furthermore, to fit the sample set of scattered data points accurately, they are reoriented in the normal direction. We also observe the effect of noise in the reconstruction of bicubic B-spline surface. Experimental results demonstrate that the scattered data points are better fitted after the modification of the algorithm and the accuracy of the approximated bicubic B-spline surface is easily influenced by the presence of noise.

Keywords: 3D scattered data, B-spline approximation, bicubic B-spline surface, noisy data

1 Introduction

Surface fitting is known as surface approximation. This concept is similar to the regression problem where the model is the surface representation and the data are the sampled points on the surface [1]. Generally, there are few types of surface representation namely polygonal mesh, implicit surface, parametric surface and subdivision surface.

In the real world, surface is modelled from the large amount of scattered data points instead of uniformly distributed data points. The sources of scattered data points can be obtained from measured values of physical quantities, experimental results and computational values which are widely found in scientific and engineering applications [2]. Therefore, an efficient surface reconstruction method is needed in order to best approximate and hence to produce a smooth and accurate 3D surface from a set of scattered data points. However, the approximation for 3D scattered data points is not an easy task due to the amount of data points as well as its irregularity in distribution.

In the context of surface reconstruction, the data are usually obtained from 3D scanners in the form of point clouds. As there are many different approaches in recovering these point sets, our paper is focusing in a local area of the point sets whereas we recover the surface in patches. Using the similar approach in [3], we find a neighbourhood of points and construct a surface locally, and in the end combine the information to reform a better estimation of the whole surface.

[4] mentioned that the most frequently-used approximation methods are interpolation by spline, interpolation by radial basis function and the least square approximation. According to [5], tensor product of B-splines surfaces is widely used to approximately compare with the other types of approximation because of its advantages inherent in working with tensor products such as to provide better continuity and smoothness when involving with large data point set. A B-spline surface consists of continuous surface patches that continuously connect at their boundary and have continuous higher order derivatives. For an example, a cubic B-spline has C^2 parametric continuity which means that it is not only continuous on the knot intervals but also has continuity in tangents and curvatures. A C^2 surface has stronger continuity than continuity of curvature as C^2 ensures G^2. If it only had continuity of curvature it would be G^2. The attractive and important properties of B-spline which

* Corresponding author e-mail: kenji_liewkj@yahoo.com.my

make it stand as the most efficient surface representation such as [6]:

i. It possesses a high degree of continuity which is important for computing the surface intrinsic properties such as curvature.
ii. If subjected to an affine transformation, it is still a B-spline whose control points are obtained by subjecting the original B-spline control points to that affine transformation. Examples of affine transformation are translation, rotation, scaling, reflection and shearing.
iii. Due to the local support of the basis B-spline function, the shape can be controlled locally.

This paper focuses on the usage of B-spline approximation on a neighbourhood of points obtained from a 3D scanned data. The B-spline approximation technique is adopted from [2] as the computational time is low even if the number of scattered data points is large. According to [2], for 5,000 data points, the control points can be obtained in 0.15 seconds on a Sun SPARC10. With the current more up-to-date technology, the processing time obviously will speed up. As far as we are concerned, there is not much literature about the usage of B-spline approximation to approximate the surface from a set of scattered data points in a local setting, and also the effects of the noisy data on the B-spline surface, hence we will look into both issues in this paper. The mathematical background of B-spline, B-spline approximation algorithm and the k-nearest neighbor search method will be introduced briefly. The methodology to approximate the bicubic B-spline surface by using the modified B-spline approximation algorithm and to observe the effects of noisy data on approximated B-spline surface will be described in Section 2. Section 3 shows the graphical results of approximated bicubic B-spline surface using modified B-spline approximation algorithm and affected bicubic B-spline surface by different noise levels. Then, in Section 4, we will discuss the approximated bicubic B-spline surfaces and the relation between noise level and accuracy based on visual inspection by using the results obtained from Section 3 and finally a conclusion is given in Section 5.

2 Materials and Methods

In this paper, a set of scattered data points together with added noisy data are taken as an input and to produce tensor product B-spline surfaces as output. To fit a set of 3D data points by using B-spline surface, one can interpolate B-spline to pass through all the data points but this method will be sensitive to noise and the data may be overfitting. Therefore, one can use approximation method to produce a smoothing effect. In determining the set of control points for this paper, we will use the concept of interpolation and approximation. Before describing this algorithm, some mathematical background of B-spline surface will be provided as follows.

2.1 B-spline Surface

The rectangular B-spline surface patch $f(u,v)$ is constructed by applying tensor product technique to the B-spline curve which is described as a linear combination of B-spline basis functions in two topological parameter u and v [6]. Furthermore, it is defined by a topological rectangular set of control points $P_{i,j}$ for $0 \leq i \leq m$, $0 \leq j \leq n$ and the two knot vectors: $U = (u_0, u_1, u_2, \ldots, u_{m+k})$ and $V = (v_0, v_1, v_2, \ldots, v_{n+l})$. B-spline surface patch is given by

$$f(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{P}_{i,j} N_i^k(u) N_j^l(v) \qquad (1)$$

where $N_i^k(u)$ and $N_j^l(v)$ are the B-spline basis functions of order k and l respectively. The parameter u and v are the global parameter.

2.2 B-spline Approximation

For the discussion of B-spline approximation, we follow the literature from [2]. The materials are repeated here in detail as we want the completeness of understanding for our paper. Let Φ be a uniform tensor product grid overlaid on a rectangular domain Ω. Assume $\Omega = \{(x,y) \mid 0 \leq x < m,\ 0 \leq y < n\}$, where $m,n \in \mathbb{Z}^+$, be a rectangular domain in the xy-plane. Consider a set of scattered points, denoted as $P = \{(x_c, y_c, z_c)\}$ in 3-dimensional plane, where $(x_c, y_c) \in \Omega$ and Φ is an $(m+3) \times (n+3)$ lattice which spans the integer grid in Ω. To approximate the scattered data P, an approximation function f is formulated as a uniform bicubic B-spline function, which is defined by a control lattice Φ overlaid on domain Ω. To have a better picture, see Figure 1.

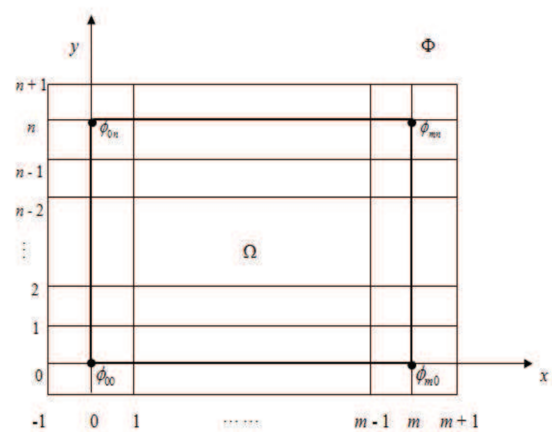

Fig. 1: Domain Ω and Lattice Φ

Let ϕ_{ij} be the value of the ij-th control point on lattice Φ, located at the position (i, j) for $i = -1, 0, 1, \ldots, m+1$

and $j=-1,0,1,\ldots,n+1$. Then, the function value of f at a position $(x,y)\in\Omega$ is as follows:

$$f(x,y)=\sum_{k=0}^{3}\sum_{l=0}^{3}N_k^4(s)N_l^4(t)\phi_{(i+k)(j+l)} \quad (2)$$

where $i=\lfloor x\rfloor-1$, $j=\lfloor y\rfloor-1$, $s=x-\lfloor x\rfloor$ and $t=y-\lfloor y\rfloor$. The parameter s and t are the global parameter. Note that $\lfloor x\rfloor$ is known as floor function and can be defined as $\lfloor x\rfloor=\max\{n\in\mathbb{Z}|n\le x\}$. N_k^4 and N_l^4 are the uniform C^2 continuous cubic (degree 3) B-spline basis functions defined as:

$$N_i^4(t)=\begin{cases}\frac{(1-t)^3}{6} & ,i=0\\ \frac{3t^3-6t^2+4}{6} & ,i=1\\ \frac{(-3t^3-3t^2+3t+1)}{6} & ,i=2\\ \frac{t^3}{6} & ,i=3\end{cases}$$

where $t\in[0,1)$. These uniform cubic B-spline basis functions serve to weigh the contribution of each control point to $f(x,y)$ based on its distance to (x,y). Therefore, the problem is reduced to determine the control points in lattice Φ that will be the best approximation for the set of the scattered points P. Let consider a data point (x_c,y_c,z_c) in P and from Equation (2), we know that $f(x_c,y_c)=z_c$ is related to the sixteen control points in the neighborhood of (x_c,y_c). To illustrate this method easily and without loss of generality, assume that $1\le x_c,y_c<2$ and thus $i=j=0$. Therefore, Equation (2) at (x_c,y_c) can be simplified as:

$$z_c=f(x_c,y_c)=\sum_{k=0}^{3}\sum_{l=0}^{3}N_k^4(s)N_l^4(t)\phi_{kl}$$

where $s=x_c-1$, $t=y_c-1$. In order to determine ϕ_{kl}, the least square sense solution is necessary to minimize the square sum of the 16 control points that is $\sum_{k=0}^{3}\sum_{l=0}^{3}\phi_{kl}^2$. The method to derive an explicit formula for the sixteen control points ϕ_{kl} can be referred to [7]. Here, we only provide a unique solution which is obtained from the derivation as the following:

$$\phi_{kl}=\frac{N_k^4(s)N_l^4(t)z_c}{\sum_{a=0}^{3}\sum_{b=0}^{3}[N_a^4(s)N_b^4(t)]^2} \quad (3)$$

where $(k,l)=(0,0),(0,1),\ldots,(3,3)$. Next, for each data points $(x_c,y_c,z_c)\in P$, equation (3) can be used to determine the set of sixteen control points in its neighborhood. These neighborhoods will overlap if the data points are close to each other.

Therefore, they may assign different values to several shared control points. For each data point, $(x_c,y_c,z_c)\in P$, Equation (3) gives a different value. For instance, let consider ϕ_c for ϕ_{ij}. The equation of ϕ_c is shown as follows:

$$\phi_c=\frac{w_c z_c}{\sum_{a=0}^{3}\sum_{b=0}^{3}[N_a^4(s)N_b^4(t)]^2}$$

where $w_c=N_k^4(s)N_l^4(t)$, $k=(i+1)-\lfloor x_c\rfloor$, $l=(j+1)-\lfloor y_c\rfloor$, $s=x_c-\lfloor x_c\rfloor$, $t=y_c-\lfloor y_c\rfloor$. ϕ_{ij} is chosen to minimize the error, $e(\phi_{ij})=\sum_c(w_c\phi_{ij}-w_c\phi_c)^2$. Differentiating $e(\phi_{ij})$ with respect to ϕ_{ij} gives

$$\phi_{ij}=\frac{\sum_c w_c^2\phi_c}{\sum_c w_c^2} \quad (4)$$

If ϕ_{ij} has contribution from several data points, then Equation (4) provides a least square solution in order to minimise local approximation error. But, if ϕ_{ij} has contribution from a data point only, then the computation is reduced to equation (3) and leaves no approximation error. Furthermore, if ϕ_{ij} does not have contributions from any data points, then ϕ_{ij} can be assigned as zero value or the average of coordinate-z_c's. To accelerate the computation, numerator and denominator of equation (4) can be accumulated for each control point by considering each data point in turn. The value of control point is obtained by division provided that the denominator is not zero. Null denominator indicates there are no any data points. The pseudocode for the B-spline approximation algorithm can be obtained in [2]. In our study, we use the similar approach as [2], that is we build up 4×4 control net and estimate the control points from the neighboring data points. In addition, there is no such multilevel step as we directly apply the B-spline approximation method. This is because the algorithm is simple and easy to implement. We study and modify this algorithm so that we can still achieve the similar result without undergoing the multilevel.

2.3 k-Nearest Neighbours Algorithm

The k-nearest neighbours (kNN) algorithm is one of the simplest, yet very accurate classification method [8]. It was a type of machine learning algorithm that was first mentioned and described in the early 1950s but only attracted the attention in 1960s as computing power started available [9]. Here the k is a positive integer. This algorithm searches for the points that are relatively close to a considered point from a set of points in n-dimensional space. Metric to measure the closeness is the Euclidean distance where the Euclidean distance between two $n-$dimensional points, $X=(x_1,x_2,x_3,\ldots,x_n)$ and $Y=(y_1,y_2,y_3,\ldots,y_n)$ is given as follows:

$$d(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

Since k is a user-defined value, one has to choose it carefully. If the k value is too small, noise will be modelled whereas for large k, the neighbours may include many points from other classes. This algorithm is very simple and there is no training involved but it is costly as the same searching procedure is repeated for every single points in a point set.

2.4 Modified B-spline Approximation Algorithm

In this section, we modified the approach by [2]. As in their paper, their sample is scattered but not into the context of surface reconstruction. We claim that our approach in this paper is to reconstruct a region from a 3D point sets from a real object which contains real life surface property with its feature, curve and sharp edges. The existing B-spline approximation algorithm is undergoing a modification in order to minimise the distance between the scattered data points and the approximated bicubic B-spline surface. The distance is the problem in the existing algorithm because we observe that the constructed B-spline surface does not fit closely to the data points. The sample of scattered data points is chosen from a specific region in the point set model by using k-nearest neighbour search method. Before constructing the surface patch, we determine the control points, ϕ_{ij} for the region where all the sample scattered points distributed. This will eliminate the ϕ_{ij} that does not have any contributions from data points. If we set the control net with dimension $m \times n$, then when particular ϕ_{ij} in the control net does not have any contributions from data points, the $z-$coordinate of ϕ_{ij} will be assigned as zero by default. However, we will assign another numerical for it and it will be discussed in the coming Algorithm 2.

The control points near with the border of surface patch will have smaller z-coordinate value of ϕ_{ij} or height of control point if there is less contribution from the data points. By default, the z-coordinate of control points are assigned with zero value if there is no contribution from data points. This normally can be observed near with the border of surface patch, therefore produce a faulty result as the surface move towards the zero. To solve this border problem, the height of the particular control points which is lower than the average value of all coordinate z_c, is assigned to a value of its nearest neighbouring point where that particular control point acts as the centre of the neighbourhood. Note that the value of the nearest neighbouring point is from the sample scattered data points.

Next, when the set of scattered data points is significantly below or above the B-spline surface patch, then adjustment of the control points is performed in order to fit the scattered data points better. In order to carry out the adjustment, the algorithm 1 is needed to calculate the average distance, which is a problem from the existing B-spline approximation algorithm. The algorithm is shown as follows:

Algorithm 1

Input: Scattered data points $P = \{(x_c, y_c, z_c)\}$
Output: Average distance, k

Step 1:Compute the set of unadjusted control points, $\phi_{ij} = \{(q_x, q_y, q_z)\}$ for the centre of scattered points region by using B-spline approximation algorithm.

Step 2:Generate function, $f(x,y)$ using linear interpolation from $P = \{(x_c, y_c, z_c)\}$.

Step 3:Generate n random points, $P_r = \{(x_r, y_r, z_r)\}$ from bicubic B-spline function, $P(u,v)$ based on ϕ_{ij}. (*n is an user-defined value and must be large enough*)

Step 4:Compute $z = f(x_r, y_r)$ by substituting value of x_r and y_r.

Step 5:Compute $d = z_r - z$ and then the average distance, $k = d/n$.

Before constructing bicubic B-spline surface patch, the position of the sample scattered data points, P needs to be reoriented in order to have a better surface approximation in order to increase the accuracy of the fitting. Principal component analysis (PCA) is a statistical procedure, which is used to reorient the data points to the orthogonal direction. Therefore, this will increase the accuracy of the fitting as mentioned. In this research, the normal vector of P is estimated by the PCA. A normal vector can be obtained by determining the eigenvector for the smallest eigenvalue, which is derived from a covariance matrix. We follow the literature from [10]. Given a neighbourhood of 3D points $\{p_0, p_1, p_2, \ldots, p_n\}$ and let say p_0 is the centre of the neighbourhood. A 3×3 covariance matrix, b_{ij} where $i, j = 1,2,3$ is constructed such that:

$$b_{ij} = \sum_{k=1}^{n} (p_{k_i} - p_{0_i})(p_{k_j} - p_{0_j}) \tag{5}$$

where $p_k = (p_x, p_y, p_z)$ is the neighboring point, and i in p_{k_i} is the index for p_k. For further explanation, p_{k_1} is the x-coordinate of p_k, p_{k_2} is the y-coordinate of p_k, and p_{k_3} is the z-coordinate of p_k. The same explanation is applied for p_{k_j}, p_{0_i}, and p_{0_j}. Let $a \le b \le c$ be the three eigenvalue of the matrix in Equation (5), the eigenvector corresponding to the smallest eigenvalue is the normal vector at p_0. Let say, the eigenvectors for the eigenvalues a, b and c are (n_1, n_2, n_3), (y_1, y_2, y_3) and (x_1, x_2, x_3) respectively. These eigenvectors are written in matrix form below:

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & n_1 \\ x_2 & y_2 & n_2 \\ x_3 & y_3 & n_3 \end{pmatrix}$$

In fact, matrix $\mathbf{A}$ is an orthogonal matrix because it is a square matrix and satisfying the condition such that $\mathbf{A}^{-1} = \mathbf{A}^T$. The rows and columns of an orthogonal matrix are an orthonormal basis which mean that each row and column have length one and are mutually perpendicular. Next, we would like to map the orthornomal system $(\mathbf{X}', \mathbf{Y}', \mathbf{N}')$ which is centered at p_0 to Cartesian coordinate system $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ which is centered at origin $(0,0,0)$. In order to do so, a transformation matrix $\mathbf{M}$ is needed for the orientation part as described in [11]. Let $\mathbf{X} = (1,0,0)$, $\mathbf{Y} = (0,1,0)$, $\mathbf{Z} = (0,0,1)$, $\mathbf{X}' = (x_1, x_2, x_3)$, $\mathbf{Y}' = (y_1, y_2, y_3)$ and $\mathbf{N}' = (n_1, n_2, n_3)$.

The rotation matrix **M**:

$$\mathbf{M} = \begin{pmatrix} x_1 & x_2 & x_3 & 0 \\ y_1 & y_2 & y_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and the inverse transformation is the transpose:

$$\mathbf{M}^{-1} = \mathbf{M}^T = \begin{pmatrix} x_1 & y_1 & n_1 & 0 \\ x_2 & y_2 & n_2 & 0 \\ x_3 & y_3 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

If wish to transform $\mathbf{X'Y'N'}$ which is centered at p_0 to **XYZ**, we can obtain p_i' such that

$$p_i' = \mathbf{M} \begin{pmatrix} p_{i_x} - p_{0_x} \\ p_{i_y} - p_{0_y} \\ p_{i_z} - p_{0_z} \\ 1 \end{pmatrix} \quad (6)$$

where $i = 0, 1, 2, \ldots, n$
To orientate back from **XYZ** to $\mathbf{X'Y'N'}$ which is centered at p_0, which is given by

$$p_i = \mathbf{M}^T \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i_z}' \\ 1 \end{pmatrix} + \begin{pmatrix} p_{0_x} \\ p_{0_y} \\ p_{0_z} \\ 0 \end{pmatrix} \quad (7)$$

With all this materials, we can now describe the steps to construct the bicubic B-spline surface patch from the sample of scattered data points. The steps are described as algorithm as follows:

Algorithm 2
Input: Point set model
Output: Bicubic B-spline surface patch

Step 1: Translate the point set model to the octant where coordinate x and coordinate y have positive values. This is because the modified B-spline approximation algorithm can only be used when $x-$ and $y-$coordinate of data points are in positive value, whereas the $z-$coordinate can be positive or negative.
Step 2: Use k-nearest neighbour search method to select a set of sample scattered data points, P from point set model.
Step 3: Ensure the height of coordinate z_c of P are approximately minimum t units by reducing or increasing the coordinate z_c. The t value here is depending on the sparseness of point set. In this research, we choose $t = 5$. The significance for t units is to avoid the irregular and bumpy surface and thus improve the accuracy of the surface patch fitting using this modified algorithm.
Step 4: Translate the position P so that centre of P is moved to origin $(0,0,0)$.
Step 5: Estimate the normal vector, **N** of P at the centre of neighbourhood, p_0 by using PCA method.
Step 6: Reorient the position of P based on Equation (6). Further rotation is needed if the undesired position occurs. The undesired position is occurred because PCA sometimes does not reorients the data points towards the orthogonal direction. After the reorientation, translate back P to the original position.
Step 7: The density of P is scaled down by a value which is based on the sparseness of P. The data points are sparse and therefore scaling down them to make them compact. This step also wants to prevent the irregular and bumpy surface and thus leads to a better approximation.
Step 8: Proceed to Algorithm 1 to find k.
Step 9: Modify the value of z_c in Equation (3) such that $z_c' = z_c - \frac{k}{a}$. (*a is an user-defined value*) This step is to make a distinct different from the existing B-spline approximation algorithm. The existing algorithm cannot fit the surface closely, and therefore Algorithm 1 comes to solve the distance problem as mentioned earlier.
Step 10: Compute the set of control points, $\phi_{ij} = \{(q_x, q_y, q_z)\}$ for the scattered points region by using B-spline approximation algorithm. If the coordinate q_z of ϕ_{ij} is less than or equal to the average z_c value of P, replace the coordinate q_z of ϕ_{ij} with the coordinate z_c value of P by considering 1-nearest neighbour search. Here, the significance of average z_c value is to acts as a marker.
Step 11: Rescale the set of control points and P and then reorient to the original position by taking the dot product of $\mathbf{M}^T$.
Step 12: Bicubic B-spline surface patch is constructed based on the set of control points that are obtained from Step 11.

2.5 Effects of Noisy Data on B-spline Surface Approximation

Now, in this section, we are going to observe the effect of noisy data with different levels on the sensitivity of B-spline surface approximation. The presence of noise will contribute to the bad fitting of the surface. Moreover, the accuracy of the 3D model will be reduced during the surface reconstruction due to the set of data points being contaminated by the noise. Although many of the surface reconstruction procedures assume that the distribution of the noise is Gaussian or normal distribution, that is not the exactly the case [12].

In order to carry out this experiment, we assume that the sets of sample scattered data points, P that are used earlier are noise free. For the experimental purpose, original data points are altered that is twenty noisy data are added randomly in positive and negative direction to P and then we construct the bicubic B-spline surface using steps in Algorithm 2. The noise level to be considered is

0.3, 0.5, and 0.7. Then, the effect of noisy data is inspected visually by making a comparison between the noise free and the added noise surface.

3 Results

We test our model with Stanford bunny as shown in the coming figures. We choose a random region to test how our algorithm perform for the data points obtained in the context of surface reconstruction.

Here, we will show the results obtained from Algorithm 2. The scattered data points are denoted as green dots whereas the control points are denoted as red dots. To select the sample of scattered points, P from Bunny point set model, we set $k = 100$, that is 100-nearest neighbour. The two selected regions are red in colour, which are shown in Fig. 2.

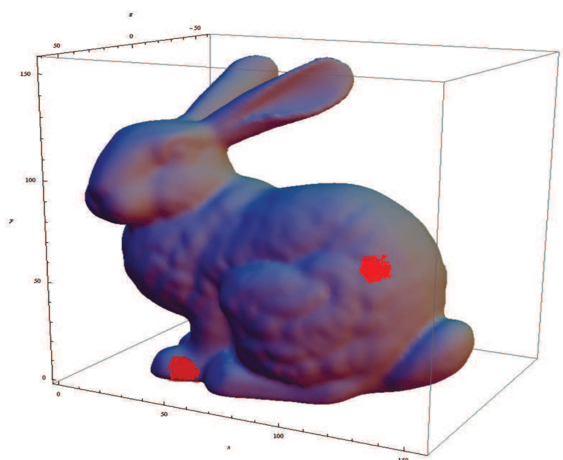

Fig. 2: The two selected red sample regions from the Stanford bunny mesh model

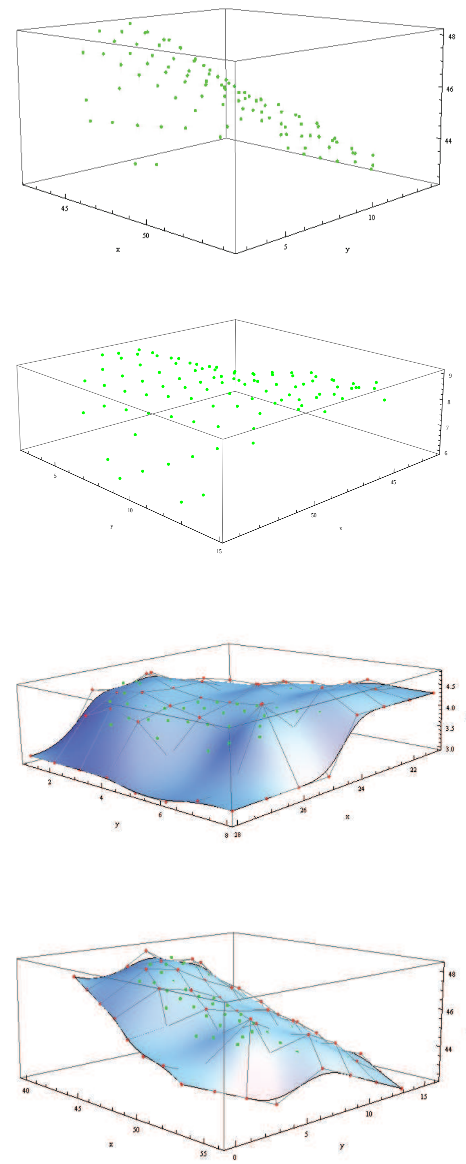

Fig. 4: Set 1: (From top to bottom) Original position of P before reorientation, position of P after reorientation, a bicubic B-spline surface patch is constructed after P undergoes reorientation, and a bicubic B-spline surface is reoriented to original position of P.

The following figures show the two different sets of P of our result.

(i) A bicubic B-spline surface patch 1

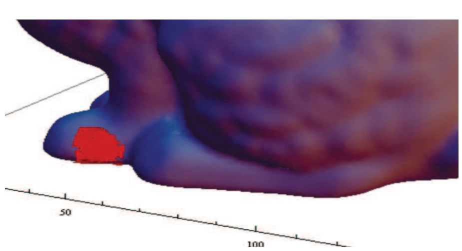

Fig. 3: The selected red sample region is zoomed

(ii) A bicubic B-spline surface patch 2

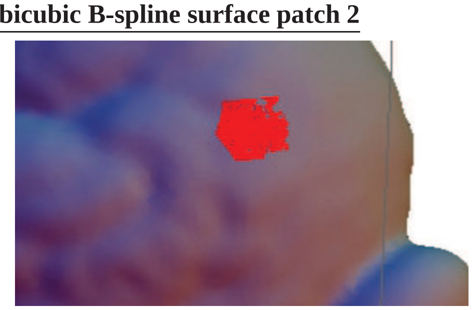

Fig. 5: The selected red sample region is zoomed

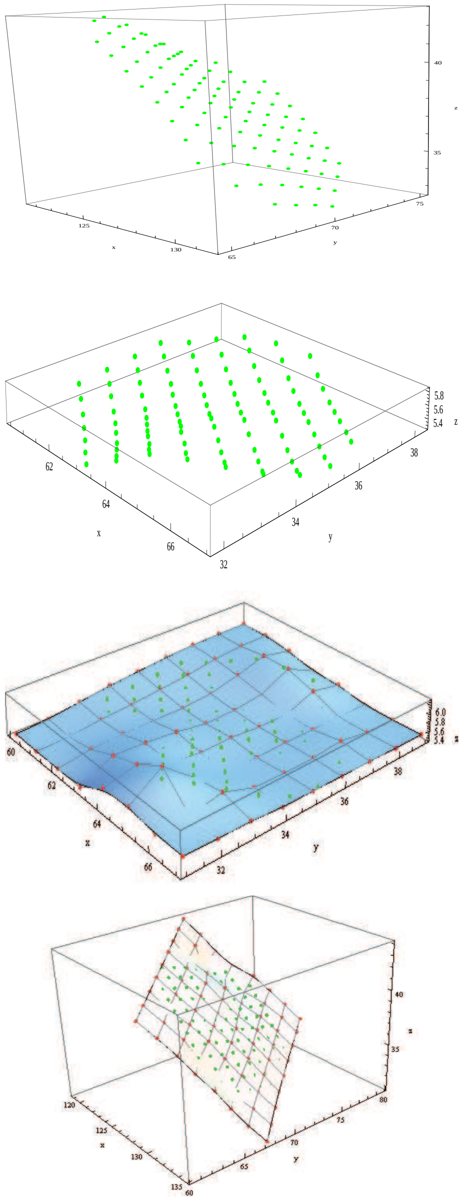

Fig. 6: Set 2: (From top to bottom) Original position of *P* before reorientation, position of *P* after reorientation, a bicubic B-spline surface patch is constructed after *P* undergoes reorientation, and a bicubic B-spline surface is reoriented to original position of *P*.

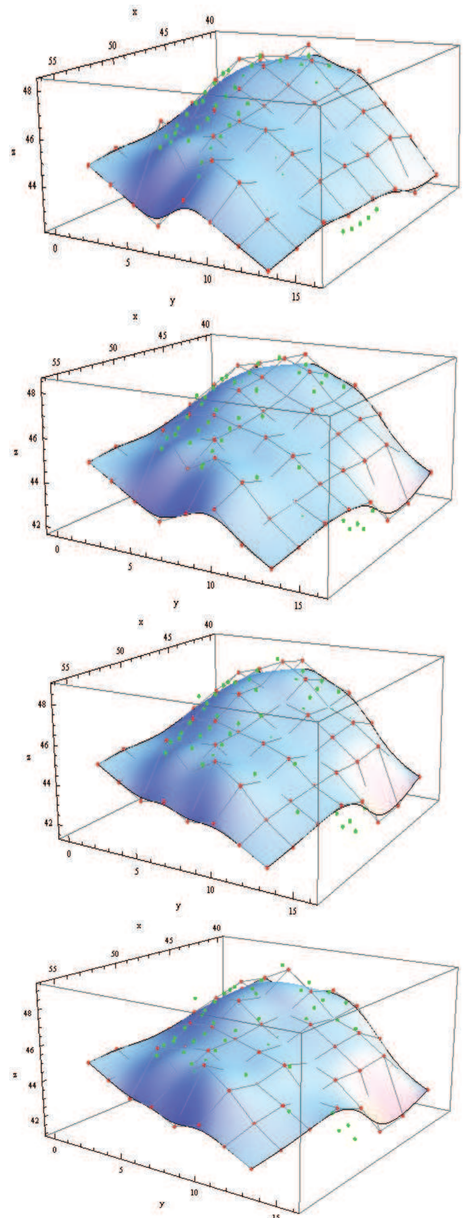

Fig. 7: Set 1: Reconstruction with no-noise, with 0.3, 0.5 and 0.7 noise level.

After looking at the surface approximation of the two different samples of scattered points, *P*, we will observe the effect of noisy data towards *P* which is assumed to be smooth. The same two samples of scattered points are used for the following experiments. Twenty noise data are added randomly in positive and negative direction in *P*. Observations for bicubic B-spline surface are as follows:

(i)Effect of noisy data towards bicubic B-spline surface patch 1

(ii)Effect of noisy data towards bicubic B-spline surface patch 2

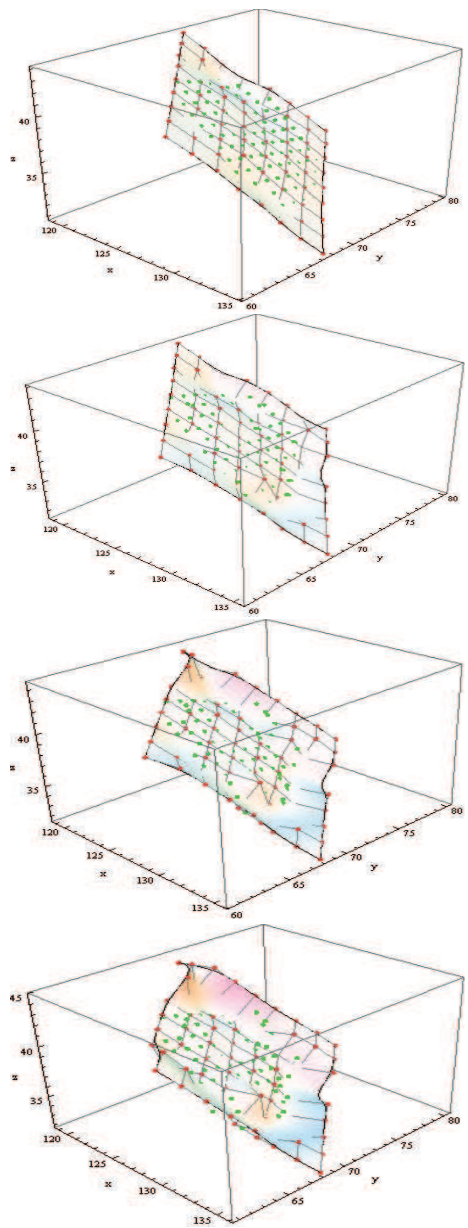

Fig. 8: Set 2: Reconstruction with no-noise, with 0.3, 0.5 and 0.7 noise level.

4 Discussion

Our bunny point set model is translated to the octant which has positive coordinate x and y, This is because we want to ensure our modified B-spline approximation algorithm can be applied in determining the set of control points. Depending on the sparseness of the scattered data points, its density can be scaled up or down in order to have a better approximation and hence to produce a smooth bicubic B-spline surface. In our experiment, the first and the second sample scattered data points are scaled down by two. Without undergoing the scaling procedure, the obtained approximated surface will be irregular and bumpy. Adjustment of the coordinate z_c of P to approximately minimum five units in the early step is done because of the higher z_c values of P, which will also contribute to the irregular and bumpy surface. However, before the construction of surface patch, the density of data points and the height of z_c values are reverted to its original values.

In the step 3 of Algorithm 1, the value n that is used for our study is 980100 points. Meanwhile, in the step 9 of Algorithm 2, we have mentioned that a is a user-defined value, and therefore the a value for bicubic B-spline surface patch 1 and 2 is 1.3 and 1.1 respectively. We believe that the value of a is related to the scaling factor and can be adjusted based on the visual inspection. Besides, we test the algorithm in a much more complicated area such as the bunny ear region. We acknowledge that our method is unable to recover this region nicely due to its complexity. However, this is a common issue when dealing with feature preservation by using surface approximation method.

The two bicubic B-spline surface patches are well-fitted with the modified B-spline approximation algorithm after resolving the distance issue in the existing algorithm as shown in Fig. 4 and Fig. 6. For the comparison purpose, we zoom in the selected region from the Stanford bunny model as shown in Fig. 3 and Fig. 5. in between the constructed bicubic B-spline surface and the original surface. We can observe that the constructed surfaces do a good approximation of the shape of the original surface. Throughout our experiment, B-spline surface with degree three is the main focus because it is not only continuous on knot intervals but also has continuity in tangent and curvature.

The effect of noisy data towards the accuracy of surface fitting is inspected visually. Here, we choose the noise level at 0.3, 0.5 and 0.7 for the experimental purpose. One is able to observe the effect of noise for the bicubic B-spline surface patch 1 via careful visual inspection because of the uneven distribution of P. It shows a slight change of the surface border when noise level is increased by observing Fig.7. However, noise effect is very obvious when the noise level is increased for bicubic B-spline surface patch 2 as shown in Fig. 8 due to the set of P being evenly distributed. Therefore, the accuracy of the approximated bicubic B-spline surfaces is quite sensitive to the presence of noise even at a low level of noise. For our experiment purpose, noise level at 0.3 is a good indication of the result as it is neither too low nor too high. Noise level at 0.5 and 0.7 can be considered quite noisy and therefore the bad fitting of the surface is expected, not due to the fact that B-spline surface fails to estimate but rather than the case of bad data points given.

5 Conclusion

This paper shows the bicubic B-spline surface approximation by using modified B-spline approximation algorithm. From the observation, the modified algorithm resolves the distance issue in the existing algorithm and fits the scattered data points efficiently. For the effects of noisy data at different noise levels towards the bicubic B-spline surface approximation, the experimental results show that the accuracy and the smoothness of B-spline surfaces are easily influenced by the presence of noise. Since bicubic B-spline surfaces are used most commonly in computer-aided geometric design, denoising process is important as the pre-process of the surface reconstruction. We note that our observation is done visually which may be prone to a subjective opinion. For future research, one can use an objective inspection such as statistical method to assess the accuracy of the approximated B-spline surface in the presence of noisy data.

Acknowledgement

The authors would like to extend their gratitude to the school of Mathematical Sciences, Universiti Sains Malaysia, Penang for the financial support from USM short term grant no. 304/PMATHS/6312136. The first author also gratefully acknowledges the generous financial support by the Ministry of Higher Education Malaysia under MyPhD scholarship.

References

[1] R. Jain, R. Kasturi, and B. G.Schunck, Curves and Surfaces, McGraw-Hill, Inc., United States, chapter 13, pp. 365-405 (1995).
[2] 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): 228-244 (1997).
[3] A. Ramli, N. A. I. M. Rosli and I. Ivrissimtzis, Bootstrap based bilateral smoothing of point sets, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2013, pp. 879-884 (2013).
[4] R. Feng, and Y. Zhang, Piecewise Bivariate Hermite Interpolations for Large Sets of Scattered Data, Journal of Applied Mathematics 2013 (2013).
[5] B. G. Lee, J. J. Lee, and J. Yoo, An efficient scattered data approximation using multilevel b-splines based on quasi-interpolants, Proceedings of the 5th International Conference on 3-D Digital Imaging and Modeling, pp. 110-117 (2005).
[6] F. S. Cohen, W. Ibrahim, and C. Pintavirooj, Ordering and Parameterizing Scattered 3D Data for B-Spline Surface Approximation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22(6): 642-648 (2000).
[7] O. Hjelle, Approximation of Scattered Data with Multilevel B-splines, Technical report, SINTEFF Applied Mathematics, Norway (2001).
[8] H. A. Guvenir, and A. Akkus, Weighted *k* nearest neighbor classification on feature projections, Proceedings of the 12-th International Symposium on Computer and Information Sciences (1997).
[9] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, Elsevier, United States, chapter 7, pp. 314-315 (2001).
[10] A. Ramli, Bootstrap Based Surface Reconstruction, PhD thesis, Durham University, United Kingdom (2012).
[11] J. D. Foley, A. V. Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, Introduction to Computer Graphics, Addison Wesley, United States, chapter 6, pp. 161-190 (1993).
[12] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, Noise Analysis and Synthesis for 3D Laser Depth Scanners, Graphical Models 71(2): 34-48 (2009).

Khang Jie Liew is a PhD student at the School of Mathematical Sciences, Universiti Sains Malaysia. He received his M.Sc. degree by research in number theory and cryptography from Universiti Sains Malaysia (Malaysia). During his master's studies, he published his research article in an international journal and a chapter in book. He also presented his research findings in international conferences. His research interests are in the areas of computer-aided geometric design and application of number theory in cryptography.

Ahmad Ramli is currently lecturing at the School of Mathematical Sciences, Universiti Sains Malaysia. He received his PhD from Durham University (United Kingdom). He is currently interested in the areas of surface reconstruction, computer-aided geometric design and bootstrap method in error estimation. During his free time, he enjoys watching movies, writing and playing Frisbee.

Ahmad Abd. Majid is an associate professor at the School of Mathematical Sciences, Universiti Sains Malaysia. He received his PhD degree in computer-aided geometric design from Universiti Sains Malaysia (Malaysia) in 1991. His research interests are in the areas of surface design, geometric modelling, CAD/CAM computer graphics and computational mathematics.