

Approximation of 3D-Parametric Functions by Bicubic B-spline Functions

M. Amirfakhrian^{a,*}

^a*Department of Mathematics, Islamic Azad University, Central Tehran Branch, Iran*

Received: 15 January 2012; Accepted: 24 March 2012.

Abstract. In this paper we propose a method to approximate a parametric 3D-function by bicubic B-spline functions

Keywords: Bicubic functions, parametric functions, B-splines.

Index to information contained in this paper

1. Introduction
2. Parametric B-splines
3. B-spline surfaces
4. The new least squares method
5. Constructing the curve
6. Constructing the surface
7. Numerical Examples
8. Conclusion

1. Introduction

Creating freeform surfaces is a challenging task even with advanced geometric modeling systems. The problem of converting the dense point sets produced by laser scanners into useful geometric models is referred to as surface reconstruction.

Parametric curves are widely used in different fields such as , computer graphics (CG), computer aided geometric design (CAGD), computed numerical control (CNC) systems [1, 2]. One basic problem in the study of parametric curves is to approximate a curve with lower degree curve segments. For a given digital curve, there exist methods to find such approximate curves efficiently [3, 4, 5, 6]. If the curve is given by explicit expressions, either parametric or implicit, these methods are still usable. However, some important geometric features such as singular points cannot be preserved. In this paper, we will focus on computing approximate surfaces which can approximate the given surface to any precision in a similar strategy for parametric curves.

The rest of this paper is organized as follows. In Section 2, we introduce the problem. some notations and preliminary of B-splines are given in Section 3. In Sections 4, 5 and 6, we give represent the least squares method for constructing

*Corresponding author. Email: amirfakhrian@ijm2c.com

curves and surfaces. In Section 7, there are some examples which used to illustrate the method. In Section 8, the paper is concluded.

2. Parametric B-splines

The x , y , and z coordinates of a curve is represented in parametric form as

$$\begin{cases} x = x(t), \\ y = y(t), \\ z = z(t), \end{cases} \quad (1)$$

where the parameter t ranges over a prescribed set of values.

The underlying core of the B-spline is its basis or basis functions. The original definition of the B-spline basis functions uses the idea of divided differences and is mathematically involved. Carl de boor established in the early 1970s a recursive relationship for the B-spline basis. By applying the Leibniz theorem, de boor was able to derive the following formula

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p-1} - u_i} N_{i,p-1}(u) + \frac{u_{i+p} - u}{u_{i+p} - u_{i+1}} N_{i+1,p-1}(u), \quad (2)$$

where $N_{i,p}$ is the i -th B-spline basis function of order p , u_i is a member of non-decreasing set of real numbers also called as the knot sequence and u is the parameter variable.

This formula shows that the B-spline basis functions of an arbitrary degree can be stably evaluated as linear combinations of basis functions of a degree lower. The obvious defining feature of the basis function is the knot sequence u_i . The knot sequence is a set of non-decreasing real numbers. The variable u represents the active area of the real number line that defines the B-spline basis. It takes $p+1$ knots or p intervals to define a basis function. Since the basis functions are based on knot differences, the shape of the basis functions is only dependent on the knot spacing and not specific knot values.

Some of the properties of the B-spline basis functions are:

- The sum of the B-spline basis functions for any parameter value u within a specified interval is always equal to 1; i.e.,

$$\sum_{i=1}^p N_{i,p}(u) = 1.$$

- Each basis function is greater or equal to zero for all parameter values.
- Each basis function has only one maximum value.

There are three different methods commonly used to parametrize model curve data; uniform, chord length and centripetal. These methods are discussed below.

- **Uniform**

This is the simplest type of parametrization where the knot spacing is chosen to be identical for each interval. Typically, knot values are chosen to be successive integers:

$$u_{i+1} = u_i + 1.$$

For many cases, however, this method is too simplistic and ignores the geometry of the model data points.

- **Chord Length**

This parametrization is based on the distance between the data points. The knot spacing is proportional to the distance between the data points. Equation (3) reflects this relationship. This parametrization more accurately reflects the geometry of the data points.

$$\frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} = \frac{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}{\|\mathbf{p}_{i+2} - \mathbf{p}_{i+1}\|}, \quad (3)$$

in which

u_i is the i -th domain knot, \mathbf{p}_i is the i -th data point and i is the number of knot interval.

- **Centripetal**

This parametrization is derived from a physical analogy. It seeks to smooth out variation in the centripetal force acting on a point in motion along the curve. This requires the knot sequence to be proportional to the square root of the distance between the data points as shown in Equation (4).

$$\frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} = \left(\frac{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|}{\|\mathbf{p}_{i+2} - \mathbf{p}_{i+1}\|} \right)^{\frac{1}{2}}. \quad (4)$$

Other parametrization methods have been investigated. All these methods have certain circumstantial advantage over the others. There is a trade-off between geometrical representation and computation time. Typically, chord length parametrization results in a very good compromise. In any event, each parametrization results in a different shape of the curve.

3. B-spline surfaces

B-spline surfaces are an extension of B-spline curves. The most common kind of a B-spline surface is the tensor product surface. The surface basis functions are products of two univariate (curve) bases. The surface is a weighted sum of surface (two dimensional) basis functions. The weights are a rectangular array of control points. The following Equation (5) shows a mathematical description of the tensor product B-spline surface.

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{q}_{ij} N_{i,p}(u) N_{j,q}(v), \quad u \in [u_{p-1}, u_{n+1}] \quad v \in [v_{q-1}, v_{m+1}]. \quad (5)$$

where $S(u, v)$ is a B-spline surface as a function of two variables, \mathbf{q}_{ij} 's are control points, $N_{i,p}(u)$ is the i -th basis function of order p as a function of u , $N_{j,q}(v)$ is the j -th basis function of order q as a function of v , and u_i, v_j are elements of the two knot sequences related to the variables u and v , respectively.

For most computer aided design purposes, as in the case of the curve, $S(u, v)$ is a vector function of two parametric values u and v . A mathematical description of this relationship is shown below in Equation (6).

$$S(u, v) = \begin{pmatrix} \sum_{i=0}^n \sum_{j=0}^m \mathbf{a}_{x,ij} N_{i,p}(u) N_{j,q}(v) \\ \sum_{i=0}^n \sum_{j=0}^m \mathbf{a}_{y,ij} N_{i,p}(u) N_{j,q}(v) \\ \sum_{i=0}^n \sum_{j=0}^m \mathbf{a}_{z,ij} N_{i,p}(u) N_{j,q}(v) \end{pmatrix} \quad (6)$$

where x , y and z are coordinates in model space. The rectangular array of control points forms what is called a control net. Similar to the B-spline curve, the B-spline surface approximates the shape of the control net. Figure 1 shows a bicubic B-spline function.

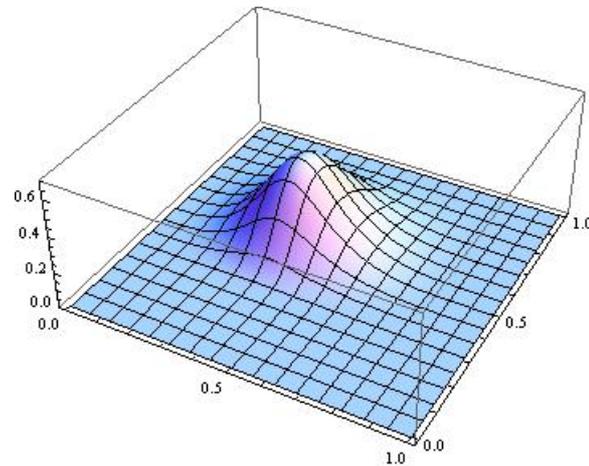


Figure 1. A bicubic B-spline function

Similar to the B-spline curve, the B-spline surface is also a network of polynomial pieces. Each piece of the B-spline surface is a two dimensionally represented part of a surface or patch. As with a B-spline curve, each patch of a B-spline surface may be represented by a periodic relationship provided the knot spacing is uniform in each direction. This is a uniform B-spline surface.

If the knot sequences are not uniformly spaced, then the surface is non-uniform. The basis functions would then have to be evaluated by the recursive relationship. The nonuniform patch Equation (5) can be represented in matrix form.

4. The new least squares method

Given a knot vector

$$\mathbf{U} = \{u_0, u_1, u_2, \dots, u_r\}, \quad u_0 \leq u_1 \leq \dots \leq u_r,$$

the associated B-spline functions $N_{i,p}$ are defined as follows:

$$N_{i,1}(u) = \begin{cases} 1, & u_i \leq u < u_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

and

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p-1} - u_i} N_{i,p-1}(u) + \frac{u_{i+p} - u}{u_{i+p} - u_{i+1}} N_{i+1,p-1}(u), \quad (8)$$

for $p \geq 2$ and $i = 0, 1, \dots, r - p$.

A B-spline curve with $n + 1$ control points is then defined as

$$C(u) = \sum_{i=0}^n \mathbf{q}_i N_{i,p}(u), \quad u \in [u_{p-1}, u_{n+1}]. \quad (9)$$

in which \mathbf{q}_i 's are control points.

With $n + 1$ data points $\mathbf{p}_0, \dots, \mathbf{p}_n$, one can find an interpolation B-spline curve. In any case, one needs to assign a location parameter τ_i to each of the data points, define a knot vector \mathbf{U} , and finally compute the control points [8, 14]. The location parameters τ_i can be assigned based on the chord length as

$$\tau_0 = 0, \quad \tau_i = \tau_{i-1} + \frac{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|}{\sum_{j=0}^n \|\mathbf{p}_j - \mathbf{p}_{j-1}\|},$$

or by using a centripetal method as

$$\tau_0 = 0, \quad \tau_i = \tau_{i-1} + \frac{\sqrt{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|}}{\sum_{j=0}^n \sqrt{\|\mathbf{p}_j - \mathbf{p}_{j-1}\|}}.$$

The knot vector \mathbf{U} can be defined as

$$u_0 = \dots = u_{p-1} = 0, \quad u_{r-p+1} = \dots = u_r = 1,$$

and

$$u_{j+p-1} = \frac{1}{p-1} \sum_{i=j}^{j+p-2} \tau_i, \quad j = 1, \dots, n - p + 1.$$

A standard interpolation problem is to solve a linear system

$$C(\tau_i) - \mathbf{p}_i = 0, \quad i = 0, \dots, n.$$

When there are $m + 1$ data points, i.e., $\{\bar{\mathbf{p}}_j\}_{j=0}^m$, with $m > n$, the corresponding location parameters $\{\bar{\tau}_j\}$ and the knot vector $\bar{\mathbf{U}}$ can also be derived from the data points $\{\bar{\mathbf{p}}_j\}$ in a similar way. Suppose that the new approximation curve corresponding to $\bar{\mathbf{U}}$ is $\bar{C}(u)$, then the least-squares method is to solve the new control points by minimizing

$$\sum_{j=0}^m \|\bar{C}(\bar{\tau}_j) - \bar{\mathbf{p}}_j\|^2.$$

Usually, the least-squares method produces well-behaved results compared to those of the standard interpolation method, but it cannot ensure that the resulting curve exactly interpolates the data points $\{\bar{\mathbf{p}}_j\}$ [3]

5. Constructing the curve

The seed curve is constructed as a cubic Bezier curve and can be written as

$$A(t) = (1-t)^3 \mathbf{q}_0 + 3(1-t)^2 t \mathbf{q}_1 + 3(1-t)t^2 \mathbf{q}_2 + t^3 \mathbf{q}_3,$$

where $\{\mathbf{q}_i\}_{i=0}^3$ are the control points. Suppose that the given curve has two end points \mathbf{p}_0 and \mathbf{p}_1 , and the corresponding tangent vectors at the end points are \mathbf{t}_0 and \mathbf{t}_1 , respectively. From the tangent constraint at the end points, we have

$$\mathbf{q}_0 = \mathbf{p}_0, \quad \mathbf{q}_1 = \mathbf{p}_0 + \alpha \mathbf{t}_0, \quad \mathbf{q}_2 = \mathbf{p}_1 - \beta \mathbf{t}_1, \quad \mathbf{q}_3 = \mathbf{p}_1.$$

When the values of α and β are determined, the corresponding cubic Bezier curve is then defined. The geometric Hermite methods such as the one in [7] can be used for determining the values of α and β . The least-squares method can also be used, which is to minimize

$$\int_0^1 \| A(t) - C(t_0 + (t_1 - t_0)t) \|^2 dt,$$

where t_0 and t_1 are the parameters of points \mathbf{p}_0 and \mathbf{p}_1 on the given curve $C(t)$, respectively. For the 2D case, we also use the inner point interpolation method, which is to select an inner point where the given curve and the approximation curve are tangent with each other. Suppose that the inner point of the given curve is $\mathbf{p}^* = (x^*, y^*)$ and $\mathbf{t}^* = (t_x^*, t_y^*)$ is the corresponding tangent vector of the given curve at \mathbf{p}^* . Let $A(t)$ be $(X(t), Y(t))$. Then we have

$$\begin{cases} X(t) - x^* = 0, \\ Y(t) - y^* = 0, \\ X'(t)t_y^* - Y'(t)t_x^* = 0. \end{cases} \quad (10)$$

The equation system (10) has three unknown variables; i.e., α , β and t , and three equations as well. The first two equations in the equation system (10) are linear with respect to α and β . The terms α and β can then be directly solved as $\alpha(t)$ and $\beta(t)$. Substituting $\alpha(t)$ and $\beta(t)$ into the third equation of the equation system (10), we obtain a univariate equation in t , which can be simplified into a univariate cubic polynomial equation $H(t)$. A brief overview of related details can be found in Appendix. By solving $H(t) = 0$, we finally obtain the values of t , α and β . Thus, the resulting approximation cubic Bezier curve is also obtained.

6. Constructing the surface

A standard interpolation problem is to solve a linear system

$$S(u_i, v_j) - \mathbf{p}_{ij} = 0, \quad i = 0, \dots, n \quad j = 0, \dots, m.$$

The least-squares method is to solve the new control points by minimizing

$$\sum_{i=0}^m \sum_{j=0}^m \|\bar{S}(u_i, v_j) - \bar{\mathbf{p}}_{ij}\|^2.$$

Usually, the least-squares method produces well-behaved results compared to those of the standard interpolation method, but it cannot ensure that the resulting curve exactly interpolates the data points $\{\mathbf{p}_{ij}\}$.

The seed surface is constructed as a bicubic Bezier surface and can be written as

$$A(t, s) = \sum_{i=0}^3 \sum_{j=0}^3 (1-t)^{3-i} t^i (1-s)^{3-j} s^j \mathbf{q}_{ij},$$

where $\{\mathbf{q}_{ij}\}_{i,j=0}^3$ are the control points. Suppose that the given surface has end points \mathbf{p}_{00} , \mathbf{p}_{10} , \mathbf{p}_{01} and \mathbf{p}_{11} , and the corresponding tangent planes at the end points are \mathbf{t}_{ij} . From the tangent constraint at the end points, we have

$$\left\{ \begin{array}{l} \mathbf{q}_{00} = \mathbf{p}_{00}, \\ \mathbf{q}_{03} = \mathbf{p}_{01}, \\ \mathbf{q}_{30} = \mathbf{p}_{10}, \\ \mathbf{q}_{33} = \mathbf{p}_{11}, \\ \mathbf{q}_{0j} = \mathbf{p}_{00} + \alpha_{0j} \mathbf{t}_{00}, \quad j = 1, 2. \\ \mathbf{q}_{3j} = \mathbf{p}_{10} - \beta_{3j} \mathbf{t}_{10}, \quad j = 1, 2. \\ \mathbf{q}_{i0} = \mathbf{p}_{00} + \alpha_{i0} \mathbf{t}_{00}, \quad i = 1, 2, \\ \mathbf{q}_{i3} = \mathbf{p}_{01} - \beta_{i3} \mathbf{t}_{01}, \quad i = 1, 2. \\ \mathbf{q}_{11} = \mathbf{p}_{00} + \alpha_{11} \mathbf{t}_{00}, \\ \mathbf{q}_{22} = \mathbf{p}_{11} - \beta_{22} \mathbf{t}_{11}, \\ \mathbf{q}_{12} = \mathbf{p}_{01} + \alpha_{12} \mathbf{t}_{01}, \\ \mathbf{q}_{21} = \mathbf{p}_{10} - \beta_{21} \mathbf{t}_{10}, \end{array} \right. \quad (11)$$

When the values of α_{ij} and β_{ij} are determined, the corresponding bicubic Bezier surface is then defined.

The relations(11) can be hold with a similar method to the equation system (10) for all variables.

7. Numerical Examples

In this section we present some numerical examples.

Example 7.1 By considering

$$U = \{0, \frac{1}{3}, \frac{2}{3}, 1\}, \quad V = \{0, \frac{1}{2}, 1\},$$

and the data points

$$\mathbf{P} = \begin{bmatrix} (0, 0, 0) & (0, 1, 2) & (0, 2, 1) \\ (1, 0, 1) & (1, 1, 2) & (1, 2, 3) \\ (2, 0, 1) & (2, 1, 2) & (2, 2, 3) \\ (3, 0, 2) & (3, 1, 1) & (3, 2, 4) \end{bmatrix}$$

we have a surface with the shape were shown in Figure 2. The flatted shape is

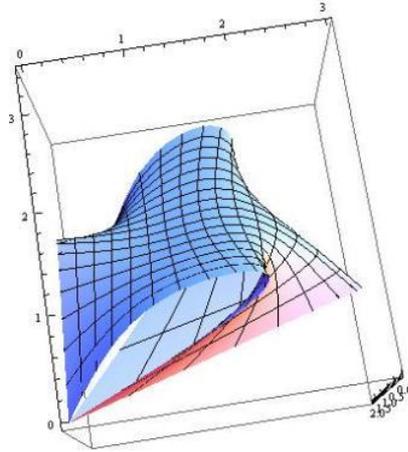


Figure 2. parametric approximation

shown in Figure 3:

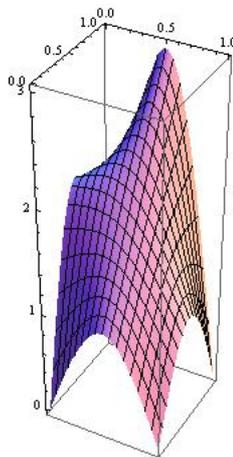


Figure 3. The flatted parametric approximation

Example 7.2 By considering

$$U = \{0, \frac{1}{2}, 1\}, \quad V = \{0, \frac{1}{2}, 1\}$$

and

$$\mathbf{P} = \begin{bmatrix} (0, 0, 0) & (0, 1, 2) & (0, 2, 1) \\ (1, 0, 1) & (1, 1, 2) & (1, 2, 3) \\ (3, 0, 2) & (3, 1, 1) & (3, 2, 4) \end{bmatrix}$$

we have a surface with the shape were shown in Figure 4.

Example 7.3 By considering

$$U = \{0, \frac{1}{2}, 1\}, \quad V = \{0, \frac{1}{2}, 1\},$$

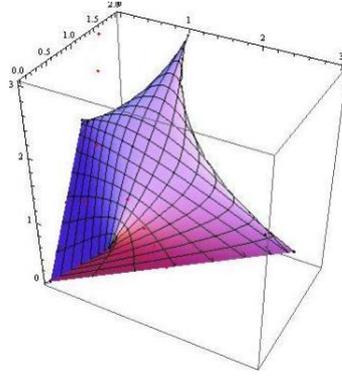


Figure 4. parametric approximation

and the data points

$$\mathbf{P} = \begin{bmatrix} (0, 0, 0) & (0, 1, 2) & (0, 2, 1) \\ (1, 0, 1) & (1, -1, 2) & (1, 2, 3) \\ (3, 0, 2) & (3, 1, 1) & (3, 2, 4) \end{bmatrix}$$

we have a surface with the shape were shown in Figure 5.

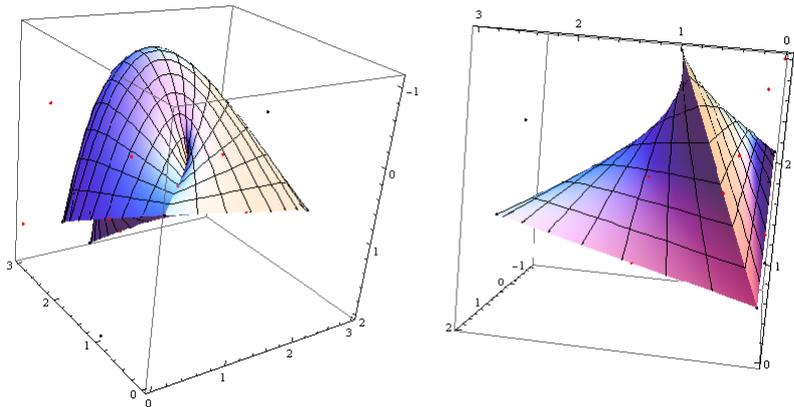


Figure 5. parametric approximation

8. Conclusion

In this work we extend a method of approximating curves by least squares to compute a surface by a bicubic B -spline functions.

9. Acknowledgement

This work was supported by Islamic Azad University, Central Tehran Branch (IAUCTB).

References

- [1] Amirfakhrian, M. and Didab, S., Approximation of Parametric Functions by Bicubic B-spline Functions, *Journal of American Scinces*, In press.
- [2] Bajaj, C., Bernardini, F. and Xu G., Automatic reconstruction of surfaces and scalar fields from 3D scans, *Computer Graphics (SIGGRAPH 95 Proceedings)* (1995), 109-118.
- [3] Chen, X. D., Maa, W. and Pau, J. C., Cubic B-spline curve approximation by curve unclamping, *Computer-Aided Design* 42 (2010) 523-534.
- [4] Curless, B. and Levoy, M., A volumetric method for building complex models from range images. *Computer Graphics (SIGGRAPH 96 Proceedings)* (1996), 303-312.
- [5] C. De Boor, *A Practical Guide to Splines*, New York, Springer-Verlag, 1978.
- [6] Fang, L. and Gossard, D., Multidimensional curve fitting to unorganized data points by nonlinear minimization. *Computer-Aided Design*, 27 (1) (1995) 48-58.
- [7] Höllig, K. and Koch, J., Geometric Hermite interpolation. *Computer Aided Geometric Design*, 12 (6) (1995) 567-80.
- [8] Hoschek, J. and Lasser, D., *Fundamentals of computer aided geometric design*. London: AK Peters; 1993.
- [9] Laurent-Gengoux, P. and Mekhilef, M., Optimization of a NURBS representation. *Computer-Aided Design*, 25 (11) (1993) 699-710.
- [10] Wang, L., Penalized least squares approximation, Intergraph design document, 2011.
- [11] Lyche, T. and Morken, K., A data-reduction strategy for splines with applications to the approximation of functions and data. *IMA Journal of Numerical Analysis*, 8 (1988) 185-208.
- [12] Meek, D., Ong, B. and Walton, D., Constrained interpolation with rational cubics. *Computer Aided Geometric Design*, 20 (3) (2003) 253-75.
- [13] Park, H. , An error-bounded approximate method for representing planar curves in B-splines. *Computer Aided Geometric Design*, 21(5) (2004) 479-497.
- [14] Piegl, L. and Tiller, W., *The NURBS book*. Second ed. NewYork, Springer-Verlag, 1997.