Thermal Image Approximation Using B-Spline Surfaces

Robert Świta¹ · Zbigniew Suszyński¹

Received: 15 February 2018 / Accepted: 14 September 2018 / Published online: 24 September 2018
© The Author(s) 2018

Abstract

This article presents a method of filtering thermal images by means of approximation using cubic B-spline patch surfaces. Approximation uses chord length parameterization with non-uniform distribution of knots, depending on the change rate of pixel values in rows and columns of the image. Advantages of approximation with such parameterization are compared with popular uniform parameterization. Presented method is very versatile and can be used for example in 3D surface topography of objects tested with stereoscopic microscopes, in resolution change of arbitrary images or modeling processes depending on two parameters. Filtering results are presented using plugin for the Soft4Image program, which was designed to unify different digital signal processing algorithms.

Keywords 3D · B-splines · Image approximation · Non-uniform parameterization · Topography

1 Introduction

Many data and measurement points are acquired in the form of grids and images. For filtering data in such domain, surface approximation with splines is very attractive. Splines have local control ability, surfaces are limited by control point's convex hull, and they are easy to differentiate. In recent years, a number of works were devoted to approximation using different kinds of B-splines: classical B-splines [1, 2], triangular B-splines [3], T-splines [4, 5], manifold splines [6], box and simplex splines [7, 8] and others. The NURBS (non-uniform rational B-splines) curves and surfaces have gained considerable popularity in graphics [9]. By adding weights to base functions, they are even wider class of splines, allowing modeling all sections of the cone (ellipses, hyperbolas and parabolas). Quality of approximation depends on surface definition, placement of knots, control points and corresponding to them,

✉ Robert Świta
robert.swita@wp.pl

¹ Department of Electronic and Informatics, Koszalin University of Technology, Koszalin, Poland

parameter values. Chordal parameter values give fourth approximation order for cubic interpolation, but for higher degrees, more accurate parameterizations, for example with arc length of cubic polynomial between approximated points, give better results [10]. Approximation can also be improved upon when using as an estimation for curve length not segment lengths between points, but results from previous iteration of approximation attempt.

The common approach to B-spline surface fitting is using least squares or weighted least squares approximation. If the parameters are computed from the data points and the knots positions from these parameters, problem can be solved using system of linear equations [1]. Many researches, in order to achieve better approximation, treat parameters, knots and control points as variables. This leads to a nonlinear least squares problem, and authors are forced to utilize more sophisticated but also more computationally demanding, iterative algorithms. Ülker and Arslan [11] developed method based on artificial immune system, Xie and Qin used conjugate gradient method [12], and Gálvez and Iglesias utilized particle swarm optimization in [13]. Many authors focus on adaptive knot placement [14–17]. Park and Lee proposed an approach that selects dominant parameter points to generate knots for B-spline curve fitting [18] and dominant columns for surfaces approximation [19].

Methods of artificial intelligence are more often used in problems of approximating irregular cloud of points, where triangulation of data (usually Delunay's) is needed.

2 B-Splines

Functions (mappings) require ability to design an inverse function, hence the need for unique assignment between argument and function value. In general, curves cannot be described as functions. However, one can represent a curve as a set of parametric functions that define coordinates of the curve points in relation to one variable t :

$$C(t) = C[x(t), y(t), z(t), \dots] \quad (1)$$

The values of the t parameter for consecutive points C_i are non-decreasing and can be interpreted as the time after which the object moving along the curve reaches given point.

The B-spline curves are a parametrically described, linear combination of the b base functions N_i^d , which are piecewise polynomials of degree d . Control points P_i , known also as de Boor points, are coefficients of these combinations. Thanks to the basic functions, it is also possible to present discrete curve points in a simple matrix form:

$$C(t) = \sum_{i=0}^{b-1} P_i N_i^d(t) \equiv C = NP^T \quad (2)$$

Base functions have limited support and create unity partition:

$$\sum_{i=0}^{b-1} N_i^d(t) = 1 \quad (3)$$

Support is also compact. Outside the local range, basis functions are equal to zero:

$$t \notin [t_i^*, t_{i+d}^*) \Rightarrow N_i^d(t) = 0$$

The parameters t^* defining the scope of the base functions are called knots. Since each base function is defined in d of the previous knots and one new, knots define connection points of the polynomials forming spline. Because of sharing the knots by the base functions, the vector t^* has the size of $b + d + 1$. Compact support determines local influence of the position of de Boor point P_i on a curve in the interval $[t_i^*, t_{i+d+1}^*)$. It also means that the position of the curve point depends only on $d + 1$ control points. Due to the limited support of the base functions, curve is contained in a convex hull of $d + 1$ control points. The sum of $d + 1$ weighted base functions is sufficient to determine the position of subsequent points on the curve (de Boor algorithm).

Since the base functions are also splines, their natural definition is the following recursive formula [20]:

$$\begin{aligned} N_i^0 &= 1 \\ N_i^d &= \alpha_i N_i^{d-1} + (1 - \alpha_{i+1}) N_{i+1}^{d-1} \end{aligned} \quad (4)$$

remembering that

$$N_i^d(t) = 0, t \notin [t_i^*, t_{i+d+1}^*)$$

and where α is a linear function of the parameter t , dependent on the knots of the base function:

$$\alpha_i = \frac{t - t_i^*}{t_{i+d}^* - t_i^*}$$

Thus, d is the degree of polynomials forming the basis functions. Shape of the curve is largely determined by the parameterization, that is, the number and distribution of the knots. Commonly used parameterizations are:

- uniform—parameters of consecutive points C are equidistant and take values of indices:

$$t_i = i \quad (5)$$

Fig. 1 B-spline basis functions for degrees $d = 1 \dots 4$ (uniform)

- chordal parameterization—the distance between parameters is proportional to the length of the curve between points C . In practice, curve is estimated with straight lines between points C :

$$t_i = \sum_{k=1}^i \|C_k - C_{k-1}\| \quad (6)$$

- centripetal parameterization—defined similarly to chordal, but the distance is measured as a square root of the Euclidean distance. With interpolation, it does not create loops and cusps, running closer to the control polygon.

$$t_i = \sqrt{\sum_{k=1}^i \|C_k - C_{k-1}\|^2} \quad (7)$$

From the approximation point of view, chordal and centripetal parameterizations are preferred. However, they are not scale invariant. Figure 1 presents basis functions for degrees $d = 1 \dots 4$ and uniform parameterization.

In the knot, B-spline can have continuity C^{d-1} , i.e., all derivatives of the order of less than $d - 1$ are continuous. For k of the same knots, the continuity of the curve decreases by $k - 1$ and the curve in knot has C^{d-k} class. Such knot is called k -multiple. By inserting a new knot or increasing multiplicity of an existing knot, one can further modify the curve. This property is used in graphics when creating curves and surfaces. Different strategies are used for the placement of knots in the problem of approximation and interpolation [21, 22]. Based on the parameters t , positions t_i^* of the curve knots are determined. Knots $t_0^* \dots t_d^* = t_0$ and $t_b^* \dots t_{b+d}^* = t_{n-1}$ are boundary knots. It is convenient to assume that $t_0 = 0$ and $t_{n-1} = n - 1$, then for uniform parameterization $t_i = i$. Internal knots, for $i = 1 \dots i_{max}$, where $i_{max} = b - d$ we distributed evenly in the index parameter domain by determining their fractional index:

$$frac = (n - 1) \times \frac{i}{i_{max}} \quad (\text{II.8})$$

Their values were determined by the linear interpolation of the parameters with indexes between which the fractional index of the knot was found.

$$t_{d+i}^* = t_{idx} + (frac - idx) * (t_{idx+1} - t_{idx}) \quad (\text{II.9})$$

where idx is an index, for which:

$$idx < frac < idx + 1$$

In this way, the choice of parameterization for the placement of knots on the curve is preserved, e.g., for chordal parameterization, knots will be distributed evenly along the curve. This parameterization guarantees that every knot span contains at least one parameter t_i (Schoenberg–Whitney condition) and covariance matrix $N^T N$ is positive definite and well-conditioned [20]. The same strategy can be used to determine curve points between approximated points after evaluation of control points (for example to draw the curve). Also, for the approximation of the function $f(x)$ using chordal parameterization, it is often important to maintain even quantized x -coordinate values. In case where no resolution change is made, this strategy ensures that the curve points are evaluated for exactly the same parameters as data points.

Approximation of function points $\mathbf{C}$ with B-spline curves consists of finding best solution to the system of linear equations in the form of a d -diagonal (banded) matrix $\mathbf{N}$ and determination of control points P_i .

$$\begin{bmatrix} N_0^d(t_0) & \dots & N_{b-1}^d(t_0) \\ \dots & \dots & \dots \\ N_0^d(t_{n-1}) & \dots & N_{b-1}^d(t_{n-1}) \end{bmatrix} \begin{bmatrix} P_0 \\ \dots \\ P_{b-1} \end{bmatrix} = \begin{bmatrix} C_0 \\ \dots \\ C_{n-1} \end{bmatrix} \quad (10)$$

For parametric curves, we can use matrix notation to determine coordinates of the control points for all dimensions at the same time:

$$N \begin{bmatrix} P_0 \cdot x & P_0 \cdot y & \dots \\ \dots & \dots & \dots \\ P_{b-1} \cdot x & P_{b-1} \cdot y & \dots \end{bmatrix} = \begin{bmatrix} C_0 \cdot x & C_0 \cdot y & \dots \\ \dots & \dots & \dots \\ C_{n-1} \cdot x & C_{n-1} \cdot y & \dots \end{bmatrix} \quad (11)$$

Each row of matrix C defines data point to be approximated. Lengths of vectors between these points are approximately proportional to the length of the curve between them and allow estimating values of the parameters t for the approximated points C in chordal parameterization.

To solve the system and obtain least squares approximation, Eq. 11 is multiplied by the transposition of matrix N :

$$N^T N P = N^T C$$

$$P = (N^T N)^{-1} N^T C = N^+ C \quad (12)$$

In practice, due to the smaller number of C columns, the left pseudoinverse of the matrix N is rarely determined. Denoting the square and symmetric covariance matrix of the base functions as N_A , and modified data vector as C_A :

$$N_A = N^T N$$

$$C_A = N^T C$$

control points approximating the curve can be evaluated by solving system of linear equations:

$$P = N_A^{-1} C_A \quad (13)$$

Worth noting is the large computational cost of the covariance matrix $N^T N$ for the number of basis functions close to the number of approximated points (when approaching interpolation). Calculations can be accelerated twice using symmetry of the covariance matrix.

In order to provide a constant derivative at the ends of the curves in the u and v directions (second derivative equal to zero), boundary conditions, called Neumann or natural conditions were added to the system of approximation equations, which provided transition of ending curves into straight lines and allowed constructing of so-called natural curves. Since the derivative of the base function of degree d is a $d - 1$ degree curve, it is easy to determine the boundary conditions:

$$N_i^{d'} = \frac{d}{t_{i+d}^* - t_i^*} N_i^{d-1} - \frac{d}{t_{i+d+1}^* - t_{i+1}^*} N_{i+1}^{d-1}$$

$$C'(t) = \sum_{i=0}^{N-1} P_i N_i^{d'} = \sum_{i=0}^{N-2} \frac{d}{t_{i+d+1}^* - t_{i+1}^*} (P_{i+1} - P_i) N_{i+1}^{d-1} \quad (14)$$

For approximation created from b cubic basis functions ($d=3$):

$$\begin{bmatrix} N_0^d(t_0) & \dots & N_{b-1}^d(t_0) \\ \dots & \dots & \dots \\ N_0^d(t_{n-1}) & \dots & N_{b-1}^d(t_{n-1}) \\ -d & d & \dots & 0 & 0 \\ 0 & 0 & \dots & -d & d \end{bmatrix} \begin{bmatrix} P_0 \\ \dots \\ P_{b-1} \end{bmatrix} = \begin{bmatrix} C_0 \\ \dots \\ C_{n-1} \\ (C_1 - C_0)(t_{d+1}^* - t_1^*) \\ (C_{n-1} - C_{n-2})(t_{b+d}^* - t_b^*) \end{bmatrix} \quad (15)$$

3 Global Surface Approximation

For data in rectangular grid form, we can utilize parametric curve approximation theory for creating surfaces described as a tensor product of the B-spline curves. For each row and column of the grid, the knot vectors are the same, but the weights of basis function (e.g., control points) are different. Given a grid of $m \times n$ data points $\widehat{S}$, approximation problem can be solved by finding a B-spline surface of degree d , defined by b control points and two knots vectors in direction u and v that approximates the data. Basis count and their polynomial degree may be different for directions u and v , but for simplicity we assume they are the same (basis are usually cubic because of higher degree polynomial's oscillations). Change of single data point position will have effect on the entire surface, so it's a global surface approximation problem. The tensor B-spline surface $S(u, v)$ is a B-spline curve in u and v cross sections:

$$\begin{aligned} S(u, v) &= \sum_{i=0}^{b-1} \left[\sum_{j=0}^{b-1} P_{i,j} N_j^d(v) \right] N_i^d(u) \equiv S = N_u P N_v^T \\ S(u, v) &= \sum_{i=0}^{b-1} C_i(v) N_i^d(u) \equiv S = N_u C^T \\ C_i(v) &= \sum_{j=0}^{b-1} P_{i,j} N_j^d(v) \equiv C = N_v P^T \end{aligned} \quad (16)$$

As apparent from (16), the problem of least square approximation of the surface described as a tensor product of the B-spline curves can be solved by two 1D approximation of curves: first approximation calculates weights C^T , second—weights P^T .

$$\begin{aligned} C^T &= N_u^\dagger \widehat{S} \\ P^T &= N_v^\dagger C \end{aligned} \quad (17)$$

Resulting surface is therefore calculated as:

$$S = N_u P N_v^T = N_u N_u^\dagger \widehat{S} N_v N_v^\dagger \quad (18)$$

Surface created from basis functions in the u direction with de Boor's points lying on the B-spline curve in the v direction would give the same surface that one obtained from base functions in the v direction with control points lying on the curve in the u direction if not for different knot parameterizations in directions u and v (the factors in (16) are interchangeable). A common approach for parameterization of regular grids, like images, is to use uniform parameterization. This can easily lead to overshooting problems and lower approximation quality. Uniform curve parameterization provides always only second order of accuracy, while chordal parameterization gives degree + 1 approximation order for degrees less than 4 [10]. In our implementation, we treated rows and columns of data as points in multi-dimensional space and calculated chordal parameterization using lengths of segments between these points. This approach is consistent with calculating data point's parameters for parametric curves from rows of matrix C as in (11).

The popular method of finding suboptimal solution to least square surface approximation consists of B-spline data approximation in v direction, followed by approximation in u direction. Changing of approximation order influences resulting surface. For example, the order column-rows give approximation S in the form:

Fig. 2 Example of image approximation with surface created by tensor product of cubic B-splines for image resolution 128×128 and number of basis functions b in u and v directions equal to: (a) 20, (b) 50, (c) 75 (d) 100

$$\begin{aligned}
 S_{cols} &= N_u N_u^\dagger \hat{S} \\
 S^T &= N_v N_v^\dagger S_{cols}^T \\
 S &= N_u N_u^\dagger \hat{S} N_v N_v^\dagger
 \end{aligned} \quad (19)$$

Expressions (18) and (19) for the final surface are identical, but matrices N_v differ in both cases, because they depend on parameterization of matrix C columns and S_{cols} rows, respectively. Also, computational cost of tensor surface is lower, because matrix C is smaller ($n \times b$) than S_{cols} ($m \times n$).

The assumption of the same knots positions for all rows and similarly for columns is very strong and affects greatly quality of approximation. The process referred as lofting (term shared from ship's hull designers) consists of constructing parallel curves approximating each column of the grid. Each column is approximated using its own parameters and knot vector. Family of these curves create approximated image in the column direction. After approximating columns, surface is processed in the same man-

Fig. 3 3D view of processed image (a) original, (b) approximation for $b = 50$

Table 1 RMS errors of surface approximation

Basis count b	RMS Error ($1E-3$)				
	Uniform param.	Tensor product	Columns-rows	Rows-columns	Lofting
5	62.4	61.1	61.6	60.7	53.6
10	55.8	49.4	49.1	45.0	33.3
15	44.0	33.6	33.3	30.3	18.8
20	35.7	26.0	25.7	23.9	15.4
25	30.3	20.0	19.9	18.9	13.6
30	25.6	15.4	15.4	14.9	12.2
50	14.0	10.6	10.7	10.7	9.5
70	9.5	8.7	8.8	8.8	7.4
90	7.6	7.4	7.5	7.5	5.7
100	6.6	6.5	6.5	6.5	4.5

ner along rows. Algorithm is simple and gives much better results, but is numerically expensive and time-consuming.

4 Results of Experiments

Thermographic images, due to lower resolution, higher background noise level and slower response time to thermal excitation, are characterized by higher noise compared to ordinary images. The SNR ratio (and consequently also the figure of merit) of IR cameras can be digitally improved by approximating such images. Such processing

Fig. 4 RMS errors of different approximation methods in function of the basis number

Table 2 Execution times

Basis count b	Time (ms)				
	Uniform param.	Tensor product	Columns–rows	Rows–columns	Lofting
5	21	23	42	43	634
10	25	27	45	48	819
15	30	31	50	51	1064
20	38	41	60	61	1413
25	44	46	67	67	1832
30	53	55	79	79	2404
50	118	120	154	156	6346
70	230	234	271	274	14 028
90	415	416	453	458	26 361
100	533	536	569	566	34 861

allows for more precise recognition of thermal inhomogeneities in the tested samples. The image presented in the analytical forms as a combination of linear bicubic surfaces has a simpler representation that allows its compression. It is independent of the original sequence resolution and simplifies its processing. An example of such processing may be the determination of thermal parameters of materials using thermal tomography algorithms, in which, instead of processing all the pixels of the sequence, only simple approximation functions are analyzed.

Figure 2 illustrates thermal image approximation with tensor B-spline surface for different number of basis functions. Images show thermal response of the silicon-eutectic-molybdenum sample of high-power thyristor structure with defects in the eutectic layer for the optical pulse excitation emitted by the flash lamp. IR sequence was recorded with a 128×128 pixels resolution and 500 FPS frequency.

Each image can be treated as a topographic map. Also vice versa, the surface map of the samples examined, e.g., by means of electron microscopes, can be filtered by approximation, improving the SNR ratio. Figure 3 presents such an interpretation of the images of the thermal response of the thyristor sample.

As all three described surface approximation methods differ only in parameterizations of columns and rows of data grid, for uniform parameterization they deliver the same results, with the approximation of tensor product surface being theoretically the fastest. In all experiments, cubic splines with normal boundary conditions in u and v directions were used. RMS errors for all methods using chordal parameterization and different number of basis in approximating exemplary image are presented in Table 1 and Fig. 4.

Computation times are given in Table 2. Execution times were measured for processing 128×128 image on i7 class processor at 2.4 GHz. All algorithms were implemented in C#, as a plugin for Soft4Image program, which was designed by

Fig. 5 Time executions for tensor, columns–rows and rows–columns methods in function of the basis number

Fig. 6 Approximation (blue) with tensor surface of the selected row (green) when row vector has C^0 class continuity (worst-case scenario) for number of basis (a) 20, (b) 50, (c) 70, (d) 100 (Color figure online)

the authors as a universal signal processing tool, especially for processing of image sequences. Images, resources and plots are also obtained using this software.

Comparison between tensor, columns–rows and rows–columns methods is presented in Fig. 5. Of course, times of columns–rows and rows–columns are nearly identical. The difference is a result of parameterization of columns vs rows. Additional calculations for the chordal parameterization also do not introduce significant changes in time execution in comparison with uniform parameterization. Lofting method has much bigger numerical complexity, and the times are incomparable with the rest of the

algorithms. It can be used when quality is the highest priority and real-time calculations are not required.

Figure 6 presents results of approximation with tensor method for selected row of image with sharp edges, prepared as a worst-case scenario in function of basis number.

5 Conclusions

This article presented three methods of image approximation using B-spline surfaces. They differ in data point and knot placement when using non-uniform parameterization. For uniform parameter distribution, all methods give the same result. This parameterization does not take into account speeds of change of the curves in u and v directions (derivatives), and the results of experiments clearly show this weakness. Much better results, with minimal overhead in computation, can be achieved using chordal parameterization. Creating surface from tensor product of B-splines proved to be the fastest method, with errors slightly worse than successive global approximation of columns and rows of the image. This method should be preferred, when approximating images. Lofting method needs improvement in reducing computational complexity, but it clearly sets the boundaries for achievable approximation errors for given number of basis functions.

Acknowledgments Materials for the article were presented at 21st European Microelectronics and Packaging Systems Conference, Warsaw 2017.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. L. Piegl, W. Tiller, *The NURBS Book* (Springer, Berlin, 2012)
2. Y. Zhang, J. Cao, Z. Chen, X. Li, X.M. Zeng, B-spline surface fitting with knot position optimization. *Comput. Graph* **58**, 73–83 (2016)
3. Y. He, H. Qin, Surface reconstruction with triangular B-splines, in *Proceedings of Geometric Modeling and Processing* (IEEE, 2004), pp. 279–287
4. J. Zheng, Y. Wang, H.S. Seah, Adaptive T-spline surface fitting to z-map models, in *Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia* (ACM, 2005), pp. 405–411
5. J. Wang, Z. Yang, L. Jin, J. Deng, F. Chen, Parallel and adaptive surface reconstruction based on implicit PHT-splines. *Comput. Aided Geom. Des.* **28**, 463–474 (2011)
6. X. Gu, Y. He, H. Qin, Manifold splines. *Graph. Models* **68**, 237–254 (2006)
7. J. Cao, X. Li, G. Wang, H. Qin, Surface reconstruction using bivariate simplex splines on Delaunay configurations. *Comput. Graph.* **33**, 341–350 (2009)
8. Y. Tan, J. Hua, H. Qin, Physically based modeling and simulation with dynamic spherical volumetric simplex splines. *Comput.-Aided Des.* **42**, 95–108 (2010)
9. G. Farin, *NURB Curves and Surfaces* (A. K Peters, Natick, 1999)
10. M.S. Floater, Chordal cubic spline interpolation is fourth order accurate. *IMA J. Numer. Anal.* **26**, 25–33 (2006)

11. E. Ülker, A. Arslan, Automatic knot adjustment using an artificial immune system for B-spline curve approximation. *Inf. Sci.* **179**, 1483–1494 (2009)
12. H. Xie, H. Qin, A novel optimization approach to the effective computation of NURBS knots. *Int. J. Shape Model.* **7**, 199–227 (2001)
13. A. Gálvez, A. Iglesias, Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Inf. Sci.* **192**, 174–192 (2012)
14. X. Zhao, C. Zhang, B. Yang, P. Li, Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation. *Comput.-Aided Des.* **43**, 598–604 (2011)
15. W. Li, S. Xu, G. Zhao, L.P. Goh, Adaptive knot placement in B-spline curve approximation. *Comput. Aided Des.* **37**, 791–797 (2005)
16. Y. Yuan, N. Chen, S. Zhou, Adaptive B-spline knot selection using multi-resolution basis set. *IIE Trans.* **45**, 1263–1277 (2013)
17. X. He, L. Shen, Z. Shen, A data-adaptive knot selection scheme for fitting splines. *IEEE Signal Proc. Lett.* **8**, 137–139 (2001)
18. H. Park, J.-H. Lee, B-spline curve fitting based on adaptive curve refinement using dominant points. *Comput. Aided Des.* **39**, 439–451 (2007)
19. H. Park, B-spline surface fitting based on adaptive knot placement using dominant columns. *Comput.-Aided Des.* **43**, 258–264 (2011)
20. C. De Boor, A practical guide to splines, in *Applied Mathematical Sciences, revised ed.*, vol. 27 (Springer, Berlin, 2001)
21. P.J. Hartley, C.J. Judd, Parametrization of Bezier type B-spline curves and surfaces. *Comput. Aided Des.* **10**, 130–134 (1978)
22. P.J. Hartley, C.J. Judd, Parametrization and shape of B-spline curves for CAD. *Comput. Aided Des.* **12**, 235–238 (1980)