A Fast Non-Uniform Knots Placement Method for B-Spline Fitting

T. Tjahjowidodo, VT. Dung, and ML. Han

Abstract—A two-step fast non-uniform knot placement algorithm applicable for noisy data is presented in this paper. The algorithm is started by evaluating the second derivative of the sampled data and, subsequently, by using the half-split approach, the best piecewise linear function is fitted to the computed derivative. In the second step, the fitted functions are adjusted to define the final knot for the B-spline fitting. The proposed method is subsequently validated on an experiment data with a known nominal surface. It is demonstrated that the proposed method offers a fast computational time that allow for online surface estimation.

I. INTRODUCTION

In data fitting applications, a knot vector must be defined in advance while the control points are identified based on the minimization of a least squared error between the data points and the fitted function. Commonly, knots are chosen in uniform space. However, this might result in an overshooting problem in the case of smooth inhomogeneous or discontinuous data. In order to overcome the problem, determining a non-uniform knot space seems to be the best solution. Unfortunately, defining the knots and their respective locations in non-uniform space is still a challenging problem as it is computationally costly ([1]).

Selection of the location of non-uniformly spaced knots is very critical on the resulting shape. Many approaches are found in the literature for optimizing the knot vector. In short, there are three main classes of approach for the knot placement algorithm. The first one is based on a trial-and-error approach. The main idea of this approach is by creating an initial knot vector and evaluating the error of the fitted curve. The error will be evaluated with a predefined stopping criterion. The computation procedure will be terminated and the selected knot vector is returned if the criterion is met, otherwise a new knot vector will be created based on a certain criterion. Although it usually offers good results, but the computational time in this approach is rather costly. Defining the knot vector can be proceeded in either ways: (i) minimum number of knots are initially defined and after quantitative checking, if necessary, an insertion of more knots is introduced to improve the fitted curve ([2],[3]); or (ii) the algorithm is started from a dense knot vector, where certain knots are eliminated if the fitted function responses with small change ([4],[5]). An evolvemental algorithm ([6],[7],[8]) and immune system ([9]) are also introduced as alternatives to the first two approaches in this trial-and-error method.

The second class in non-uniform knot placement algorithm is by utilizing an optimization process to find the optimal knot points ([10],[11],[12]). This category is rather hard to solve optimally, which is due to the nonlinearity in its cost function and the search space that might result into local optimum solution ([13]).

The knots placement in the third class is defined based on the feature(s) of the data. This approach offers a faster processing time but the optimal results cannot be guaranteed. He et al. ([14]) utilized the wavelet transform to locate the knots on the high frequency segment of the sampled function (data). Wavelet is a well-known time-frequency-representation method and it is proven to be effective in the areas of system identification and filtering ([15],[16],[17]) compared to other filtration techniques (e.g. [18],[19]). However, the processing time of the Wavelet decomposition is rather costly, and the selected knot vector is not optimized in terms of the number of knot. An alternative method was presented by Li et al. ([20]) that is based on the discrete curvature of data. In the first step, the discrete curvature of data are computed, and then a low pass filter is applied to get a smoother curvature and knots are subsequently selected by a heuristic rule. This method offers a fast processing time, but it might result in a lot of redundant knots. Therefore, this method is suggested to be used to find the initial guess for the knot vectors before further optimization processes.

Yuan et al. ([21]) proposed a multi-resolution basis function set for identifying knot location. This method is based on the fact that denser knots will occur at larger curvature positions. Initially, a subset of basis functions is selected from the pre-specified multi-resolution basis set using Lasso optimization. Subsequently, a vector space is constructed based on the basis function and a concise knot vector is identified to fit the unknown function. Despite the effectiveness of the method, the utilizing basis functions requires excessive computational time.

Conti et al. ([22]) offered an alternative to select the knot vector for a cubic spline based on the third derivative. However, this method is only applicable for smooth splines with predefined number of knots and it is not applicable for non-smoothing functions. Kang et al. ([23]) proposed another method that is also based on the third derivative of the function that is referred to as the sparse optimization. Given an initial dense set of knots, by comparing the left and right third derivatives for a cubic spline from a certain knot (which are constants), the knot will be eliminated if the difference between the two third derivatives is smaller than a certain threshold. That is to say that the second derivatives essentially are referring to collinear lines. Executing
repetitive process, the real knots will eventually survive after the process. The knot vector is also adjusted to find the best result. However, this method is not very effective for noisy data and it is not very efficient to handle multiple knots.

This paper presents a new way for calculating the knot vector of a cubic spline based on the second derivative (for cubic spline) of the data. In the first step, the half-split method with a certain error band as a criterion is applied to the second derivative of the sampled data. Subsequently, the best fitted linear piecewise functions for the computed second derivative is evaluated. The break points of the linear piecewise function are considered as candidates of knot vector. In the second step, an optimization process is employed to adjust the location of knots to handle multiple knots, which is reflected on the second derivative.

As the knots is solved analytically, the proposed method allows for the evaluation to nearly exact knots (both the number of knots and their locations) if the data is sampled from a specific spline function. At the same time, naturally, the method can handle multiple interior knots. This means, the method will give the best fitted for various types of B-spline (non-differentiable and discontinuous functions). Another important feature of this method is that it offers fast processing time that makes the method to be potential for real-time applications.

The paper consists of four sections. Section II presents the methodology to solve the knots based on the half-split method and a predefined threshold. Optimization of the knot location and multiple of knot handling is discussed in section III. Experimental validations are presented in section IV to demonstrate the effectiveness of the proposed method and some respective conclusions are drawn in section V.

II. HALF-SPLIT METHOD FOR KNOTS EVALUATION

A cubic spline is a piecewise function that is continuous up to its second derivative where each polynomial is connected to the other at a knot or a break-point. Mathematically, a $p$-degree B-spline, $S(t) = \sum_{i=0}^{n} N_{p,i}(t)P_i$, is defined on sequential knots:

$t_0 = t_1 = \ldots = t_p < t_{p+1} \leq \ldots \leq t_{m-p} < t_{m-p} = \ldots = t_m$

where the knots $t_{p+1}, t_{p+2}, \ldots, t_{m-p}$ are called interior knots, $p = m - n - 1$, with $P_i$ is $i$th point control and $N_{p,i}(t)$ is the $i$th basis function with degree of $p$ that is defined in a recursive procedure ([13]).

$$N_{0,i} = \begin{cases} 1, & \text{if } t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,j} = \frac{t - t_i}{t_{i+j} - t_i}N_{i,j}(t) + \frac{t_{i+j} - t}{t_{i+j} - t_{i+j}}N_{i+1,j-1}(t)$$

The second derivative of a cubic spline appears as linear polynomial piecewise functions, where the intersection of two adjoining piecewise functions is represented as a knot as illustrated in Figure 1. This means, if a spline has piecewise linear functions as its second derivative, the two adjoining straight lines will be connected through a break-point.

Given a data set of an unknown function (either clean or noisy), we can define its second derivative from the given data. Approximating the analyzed second derivative by linear piecewise functions, the intersection of each two adjoining straight lines and each parametric, $t$, as the interior knot of the cubic spline can be defined.

A. Data Splitting

In this approach, a subset of the second derivative sample points will be approximated as a straight line. The first key point of this approach is to split the data into a set of piecewise linear function. In order to perform it, we employ the half-split method to subdivide the data (in ascending order) as illustrated in Figure 2. At the first step, all the data is assumed to be a part of a single straight line with knot vector is $T_1 = (t_0, t_1)$.

Subsequently, we examine whether all data, indeed, belongs to a single straight line. If it is not, we split the subset $T_1 = (t_0, t_1)$ into two sub-intervals by introducing an intermediate knot that will turn the knot vector into $T_2 = (t_0, t_1, t_2)$. This knot vector, subsequently, will be re-examined whether the two subsets obtained from step 2 are originated from two lines. If not, another intermediate knot will be introduced.

At this point, let us assume that at step 3 the knot vector now is $T_3 = (t_0, t_1, t_2, t_3, t_4)$. Based on a stopping criterion/threshold (that will be discussed in section III B), the sub-interval $[t_0, t_1]$ is defined as a piecewise linear function and the three remaining intervals are not (note: in the figure, 1 indicates the data subset that does not satisfy the stopping criterion and 0 denotes the data subset that satisfies the criterion as a straight line).

![Figure 1. Typical B-spline curve](image1)

<table>
<thead>
<tr>
<th>Step</th>
<th>Figure 2. Half split method of data splitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image2" alt="Step 1" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image3" alt="Step 2" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="image4" alt="Step 3" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image5" alt="Step 4" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image6" alt="Step 5" /></td>
</tr>
</tbody>
</table>

...
As shown in Figure 3, the error band $AB$ of a data subset can be expressed as $AB = \max(y_i-ax_i) - \min(y_i-ax_i)$, where index $i$ represents the $i^{th}$ sample of the subset data.

![Figure 3. Error band of a data subset](image)

The error band, $AB$, will be compared to a control threshold, $\varepsilon$, that serves as a control factor and has to be properly chosen. If it is much smaller than the noise level, too many knots would be selected and the curve will be over-fitted. On the contrary, if the error band is too large compared to the noise level, the number of knots is insufficient and the curve will be under-fitted.

Based on a predefined control threshold, $\varepsilon$, the knots that define the data subset are identified. Figure 4 illustrates the use of the half split method and the control threshold to define the knot vector with 18 samples indexed from 0 to 17.

In the first iteration, (a), the knot vector is $[0, 17]$. When the error band is shown to be larger than the control threshold, the data subset is split and, subsequently, the knot vector is half-split (b) into $[0, 9, 17]$. The process is repeated until the last step, (d), no error band is greater than the control threshold and the repetitive process is terminated. As an outcome the knot vector now becomes $[0, 5, 8, 9, 12, 14, 17]$.

### III. Knots Optimization

Previous section presented the half-split method to define the knot vectors that is best fitted to the second derivative of the sampled data. However, in some scenarios, the method cannot be implemented directly. One case is illustrated in Figure 4c when redundant knots are identified after half-splitting process. The problem occurs when the total number of knots created from the method is larger than the actual knots, resulting in to redundant knots ($[5, 9$ and $14]$ in Figure 4c). In this case, certain actions have to be taken to eliminate the redundant knots, to join linear functions that belong to a same line and to generate new relevant knots to the joined lines.

#### A. Joining Collinear Lines

To avoid redundant knots, every two adjoining lines have to be examined whether both lines can be merged as a single line. The two lines are considered as a single subset if both error bands are smaller than a same control threshold as well as the merged line. Figure 5 illustrates the elimination of two redundant knots and joining two pairs of subset ‘[0, 5]’ to [5, 8]’ and ‘[5, 8]’ to [8, 9]’.

#### B. Knots Adjustment

In real cases, it is less likely that the identified knots coincide with the sample points. There are two possible cases where adjustment of the physical knot point to the analyzed knots locations is required. The first case occurs...
when the intersecting point of the two adjoining lines (analyzed knot) is close to a real physical knot and the neighboring physical knots [the \((i-1)^{th}\) and/or the \((i+1)^{th}\)] is more than one sample away from the corresponding physical knot. The case is illustrated in Figure 6a. In this case, the identified physical knot at \(i^{th}\) is replaced with the new analyzed knot and then, for regressing the line, the knot vector convention is converted from index number to real parametric values, \(t\).

The right panel of Figure 6 illustrates the second case when the intersecting point of the two adjoining piecewise linear lines lies between two knots \([i^{th}, (i+1)^{th}]\). Similar to the previous case, the two identified physical knots are replaced with the new knot identified from the intersecting point. For regressing purpose, the knot vector will also be represented in real parametric values. In this particular case, as the two physical knots will be replaced by a new defined knot, the total number of knot will be one number less than that in the original condition.

C. Multiple Knots

A unique feature of the B-spline technique is its ability to capture smooth and non-smooth functions. In particular, three distinct non-smooth functions, a function with a kink on its first derivative, a function with a kink, and a discontinuous function. A p-degree B-spline is continuous up to its \((p-1)^{th}\) derivative at knots. At a special knot case, when there is \(k\)-multiplicity of the knots then the B-spline function will only be continuous up to \((p-k)^{th}\) derivative.

Double knots

For a cubic spline with double knots then at the repetitive knots position the B-spline function will only be continuous up to first derivative \(y'\). This means the second derivative of the function at the corresponding point will be discontinuous. The case is illustrated in Figure 7.

There are four possible cases of double knots. In the first case, the double knots are identified close to the \((i+1)^{th}\) sample and the two physical knots are identified next to the \((i+1)^{th}\) sample, i.e. at \(i^{th}\) and \((i+2)^{th}\) sample points, while the intersecting point of the two adjoined lines lies outside the \([i, i+2]\) interval. In this case, the two physical knot locations at \(i^{th}\) and \((i+2)^{th}\) will be substituted by a double knot point at \((i+1)^{th}\). The case is illustrated in Figure 8a.

The second case (Figure 8b) occurs when two small intervals with a physical knot at \((i+1)^{th}\) is identified. A double knots will be defined at \((i+2)^{th}\) and the same treatment to the two adjacent physical knots as in the previous case will be taken.

The third case happens when the double knot is located near to \((i+1)^{th}\) or \((i+2)^{th}\). In this case, as shown in Figure 8c, the half-split process will divide the interval \([i, i+3]\) into 3 sub-interval, namely \([i, i+1]\), \([i+1, i+2]\), and \([i+2, i+3]\). In this case, the double knot is defined at the middle of sub-interval \([i+1, i+2]\), denoted by \((i+1.5)\) to substitute the four physical knots at \((i), (i+1), (i+2)\) and \((i+3)\).

The last case (Figure 8d) is similar to the third case, except that only one segment with 3 space interval (or two physical knots) is identified and the intersection of two adjoining piecewise linear functions lies outside \([i, i+3]\).

Triple knots

A B-spline function with a kink will have discontinuity at its first and second derivatives as illustrated in Figure 9. When two small interval with a center at \((i+1)^{th}\) and two physical knots adjacent to the center at \(i^{th}\) and \((i+2)^{th}\) points, the triple knots location will be chosen at the middle point of interval \([i, i+2]\) to replace three physical knots located at \((i), (i+1)\) and \((i+2)\) with the triple knots at the middle point, \((i+1)\). The case is illustrated in Figure 10a.

In case when the triple knots is identified inside \([i+1, i+2]\), as shown in Figure 10b, the half-split process will result in three single-space sub-interval \([i, i+1]\), \([i+1, i+2]\) and \([i+2, i+3]\), with a very sharp spike. The triple knots will be defined at the middle of \([i+1, i+2]\) interval to replace to four respective physical knots.

Fourfold knots

Fourfold knot will occur when we have discontinuous data as illustrated in Figure 11. The location of fourfold knot will be defined at the middle point of interval \([i, i+3]\), denoted by \((i+1.5)\). The fourfold knot, subsequently, will replace the four physical knots at \((i), (i+1), (i+2)\) and \((i+3)\).

Figure 6. Two cases of knot adjustment

Figure 7. Double case knot

Figure 8. Typical double knot cases
profile at the middle part than that at the two ends and, in turn, it will cause more noise on its second derivative.

The more irregular the profile, the higher the knots will be identified. The case occurs in the middle part of the propeller data points. Because of the rougher profile on the measurement data at the middle part, denser knots are identified at the corresponding region as indicated by triangles (▲) at the bottom part of the right panel in Figure 13. Therefore, as the entire fitting process is carried out with a uniform threshold, ɛ, the denser knots will result in an overfitted curve.

Figures 14 shows the measurement results and the fitted curve of the computer mouse with irregular clay additives. As shown on the left panel, the test sample features a few sharp ridges as indicated approximately at position 4mm and 27mm. These features correspond to the second derivative plot on the right panel of the same figure, where it shows two high spikes at the respective sharp spike locations. The proposed method is shown to be able to capture the features with higher number of knots (denoted by ▲) with no overshoot. This demonstrates the ability of the method to handle multiple knot problem.

![Propeller and computer mouse](image)

**IV. EXPERIMENTAL VALIDATION**

Validation of the proposed method is carried out on experimental data that were taken on a free form surface. In this case, we consider a twisted surface in a mini-UAV propeller and a computer mouse with some clay additives to simulate product defects. For the latter case, two different scenarios are considered, i.e. (i) with arbitrary-shaped additive, and (ii) with controlled-shape additive.

The testing apparatuses together with the sampling planes are shown in Figure 12, where the left panel shows the propeller and the right one illustrates the computer mouse with arbitrary additive shape. Keyence LJ-V7080 profilometer is used with uniform sampling space of 0.05mm to acquire the sampled points.

The left panel of Figure 13 shows the sampled profile of the propeller together with the b-spline fitted curve, while the right one depicts the calculated second derivative of the sampled data. As shown in the left panel of the figure, the proposed method is proven to be able to fit the propeller shape curve.

One thing can be highlighted from the second derivative plot is that the middle parts (between 15-30mm on the plot) indicate more noise compared to the data from the other end parts. This is due to numerical differentiation of the sampled data. The noise is attributed to the roughness surface on the propeller apparatus. During the measurement, the test sample is horizontally fixed to its normal axis. The line of sight of the noncontact profilometer is perpendicular to the normal surface of the middle part of the test sample. This orientation brings higher sensitivity to the roughness profile at the middle part of the test sample during the measurement. Consequently, the measurement data indicates rougher

![Propeller and computer mouse fitting](image)
TABLE I. RESULT SUMMARY OF THREE CASE STUDY

<table>
<thead>
<tr>
<th>Data</th>
<th>Size of data</th>
<th>Processing time (s)</th>
<th>Interior knots</th>
<th>RMSE (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1</td>
<td>801</td>
<td>0.023</td>
<td>16</td>
<td>9.477e-3</td>
</tr>
<tr>
<td>Profile 2</td>
<td>723</td>
<td>0.063</td>
<td>88</td>
<td>9.561e-3</td>
</tr>
<tr>
<td>Profile 3</td>
<td>742</td>
<td>0.039</td>
<td>32</td>
<td>1.009e-2</td>
</tr>
</tbody>
</table>

To validate the capability in handling the multiple knot case, clay additives with a regular profile is appended to the computer mouse. The clay is molded and finished carefully to get smooth surfaces and sharp edges, where the profile is shown in the left panel of Figure 15. As expected, two spikes are identified on its second derivative data that correspond to the abrupt changes of the surfaces, where denser knots are identified on the corresponding locations. The fitted curve is also shown on the left panel of the figure and no indication of overshooting implies the ability of the method to deal with a multiple knot problem. Quantitative performances of the fitted curves are tabulated in Table 1 together with quantitative measures in terms of root-mean-square error (RMSE).

V. CONCLUSION

A fast method for knots calculation in a B-spline fitting based on the second derivative is proposed. The working principle of the method is based on employing the half-split method and controlling it with a predetermined threshold to find the best fitted piecewise linear function on the computed second derivative and to identify the knots. Considering the effect of multiple knots on the second derivative, our proposed method can naturally handle multiple knots. This method is proven to be able to reconstruct a B-spline from sample data, if the second derivative of the given data can be computed with permissible error.

One particular attention has to be highlighted that the performance of the proposed method depends on the accuracy of the second derivative computation. Nevertheless, this can be easily mitigated by utilizing denser data point to improve the accuracy of the knots identification.

As the method is shown to be superior in the processing time, it is very potential in the application of online nominal surface reconstruction that is commonly required in remanufacturing processes for defect identification and feature removal. The results of the surface reconstruction utilizing the proposed method are communicated and presented in another paper.

ACKNOWLEDGMENT

The authors acknowledge the support of Nanyang Technological University and the Advanced Remanufacturing and Technology Centre (ARTC) in this research work.

REFERENCES


The processing time is measured on Intel dual core P8600, 2.5GHz, 3GB RAM under MATLAB 2013a environment.