International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.1 (2016), pp.229-240
http://dx.doi.org/10.14257/ijmue.2016.11.1.23

An Algorithm for Finding a Reasonable Smooth Curve for Any Given Data

Hee-Jue Eun[1], Suk-Jin Kim[2] and Yong-Sung Kim[3]

Department of Computer Engineering, Chonbuk National University 664-14 DuckJinDong, DuckJin-Gu, Jeonju, Korea
hjeun@jbnu.ac.kr[1], hanuri00@jbnu.ac.kr[2]
Corresponding Author, Yong-Sung Kim, yskim@jbnu.ac.kr

Abstract

This paper suggests an assignment algorithm in order to construct a reasonable Slope for any given data along with building division algorithms for generating interpolating intervals. In connection with these, finding out the Smooth Curve algorithm and generating it are essential. Finally, the fulfillment of the mathematical error analysis and numerical experience concerning with Smooth curve algorithm result in a reasonably varityting the Smooth Curve.

Keywords: *Interpolation, Divide Algorithm, Slope, Smooth Curve, error analysis, Yesonale smooth curve, Hermite, Lagrange, Spline*

1. Introduction

Smooth curves and surfaces reasonable for a set of arbitrarily given points are being popularly used in industrial and economic fields. Specifically, smooth curves are widely used in statistical areas, economic analysis, and time series that require estimation of data in a regression line equation and least squares method. Today, as computer application technology has evolved significantly, the use of smooth curves is being extended further to various areas including industries such as design drawing, speech synthesis and tracking, computer games, animation, and art media.

For a set of data arbitrarily given in two-dimensional space, a smooth curve suitable for the data can be found by using piecewise cubic interpolation. Among the equations that are categorized to be piecewise cubic interpolations, Hermite, Lagrange and Spline are generally being used.

Taking a close look at the common characteristics of interpolation, interpolation equations vary depending on how many nearest points are applied when establishing an algorithm that assigns a slope. Another characteristic is that there is limitation such that the given data can be applied to interpolation equation only if at least one variable of the data is monotonically increasing or monotonically decreasing.

Comparing equations under the piecewise cubic (P·C) interpolation, P·C Spline shows the highest usability from a perspective of smoothness, while proving that P·C Hermite has the least error limit [1,10]. Therefore, P·C Hermite is the most superior technique in finding a solution to a differential equation because P·C Hermite yields the least error. P·C Lagrange is mostly applied to integral calculus, while P·C Spline is applied to generating smooth curves [2].

This paper aims to suggest a smooth curve algorithm taking into account individual and common characteristics of P·C interpolation methods, with focus on three views as follows. Furthermore, this paper compares and investigates the proposed algorithm through mathematical error analysis and numerical example in terms of its smoothness.

The first view is to apply the piecewise cubic method in order to improve monotonicity of the given data [3].

ISSN: 1975-0080 IJMUE

Second is to set a reasonable slope by applying characteristics of curve to the given data and the domain of monotonic P·C interpolating curve in order to improve time complexity and smoothness more.

Third is to propose a divide algorithm in order to enhance smoothness of an interpolated curve since applying uniform interval might result to lack of smoothness of a curve. For a given data set, estimation of a curve model is carried out after finding a critical point and an inflection point where curvature changes of a curve. To ensure the characteristics of a critical point and an inflection point of a curve, the given range is subdivided in accordance to the inflection points. In accordance to a curving ratio of a curve constructed within subdivided range, the interval is more subdivided by the proposed division algorithm.

2. Approach of the Proposed Algorithm

As a fitting method for a curve for a given data set, piecewise cubic interpolation method is used. To ensure monotonicity, parametric piecewise cubic interpolation is applied. To set up a reasonable differential value, the characteristics of the slope in accordance to the given data, the range of slope constructed by a critical point, and the reasonable slope within the range where monotone curve can be drawn should be configured first. Lastly, the way to consider a critical point and an inflection point will be described in order to propose the division algorithm that constructs interpolation intervals.

2.1. Piecewise Cubic Interpolation

In a method of piecewise cubic (P·C) interpolation, finding g(r1), g(r2), g(r3), g(rn), for each interval [ri,ri+1] at a range of α =r1<r2<r3…rn=b is to find approximation function f which is equivalent to an arbitrary 4-order polynomial equation pi.

In terms of main difference in different P·C interpolation equations, P·C function is determined depending on how to assign a slope Si.

P·C Hermite chooses the first order derivatives Si.=g′(r), while P·C spline the second derivative Pi-1″(ri)= Pi-1″(ri) [3][8]. Apart from this, there are two other characteristics.

First, PC interpolation method is used only if at least one variable of values that pass through the given data x, y has monotonicity. In this case, the variable having monotonicity can be considered to be an independent variable. In other words, PC interpolation method is not to be applied when both variables have no monotonicity.

Second, a slope s_i is calculated by selecting the value for the first order derivatives in conjunction with the points that pass through two points and the nearest point.

As such, in P·C interpolation method, the characteristics and estimated value of curve will be different depending on how many nearest points are used, how to select the slope s_i and the order of derivatives.

2.2. Parametric Piecewise Cubic Interpolation

To apply P·C interpolation method when both variables have no monotonicity, the way to establish P·C function is to make either of two variables be monotone. For the data without monotonicity, the data itself have a turn-back point. In this case, it can be achieved only by multi-variable curve.

Therefore, smooth multi-variable curve can be created by selecting parameters that are increased through the given data, and interpolating those two separated linear functions individually.

When $t=i$ is chosen for the i-th data (x_i,y_i) [3], smooth curve for i has an interpolated function through all points (i, v_i), two single valued curves with same parameter can be created. Now, the same parameter commonly used by two curves will become an independent variable.

As described above, the separated parametric P·C functions can be calculated independently. Therefore, this method's advantage is that it can be applied to an arbitrary set of data without need of distinguishing the type of data by a user or a program.

2.3. How to Assign a Slope for the Given Data

Generally, the first order derivatives assigned to smooth curve for the given data varies depending on the position of data value adjacent to the given data. As mentioned in 2.1, the derivative function assigned to the given data value will be different depending on the interpolation equation.

In global methods such as Spline, all data values are used to calculate the first order derivatives for each data, and the second order derivatives passing through the curve is used to compute the value of the first order derivatives.

Contrary to the global method, the local method uses only some of data values adjacent to the given data in determining the first order derivatives fit for each data.

Therefore, P·C interpolation equation will be different depending on how many nearest points are considered in the process of generating the first order derivatives. The minimum amount of data required for this process can be said to be the points that reside at left or right-side of the given data.

Direct procedures to select parameter and to assign the first order derivatives to the generated curve with use of minimum amount of data are described as follows.

Since we select $t=i$ among a set of data with x and y value from i-th data (x_i,y_i), parametric PC interpolation technique can be applied by separating it into (i, v_i) and (i, v_i). For t, the first order derivatives of PC function for each variable can be obtained by selecting the point that resides at left and right side of the given data [3].

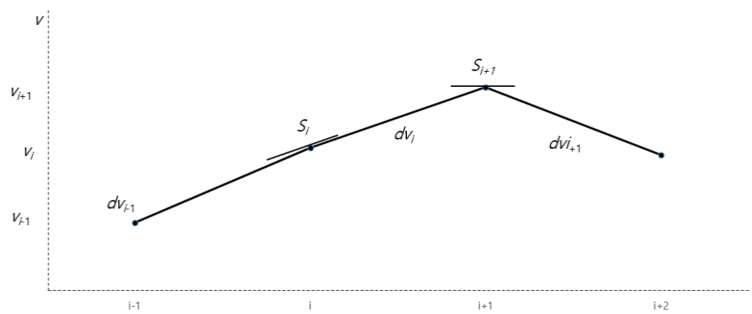


Figure 2-1. Slope Assignment Method for Any Given Data

In [Figure 2-1], if dv_{i-1} is set to be a fixed value, S is considered to be a function for dv_i. In terms of the shape for dv_i which is an independent variable, S will be drawn within the domain between two linear lines defined by y= dv_{i-1} and y= dv_i.

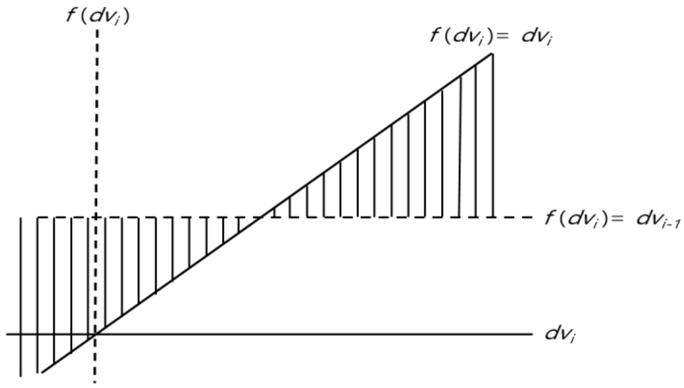


Figure 2-2. Domain of Slope

2.4. Characteristics of Curve for the Given Data

When the curve is used as a tool for representing the data, the curve should represent all the characteristics of the data within the curve. In other words, the curve should not denote unnecessary model. In case the data is presented to be the first-degree curve, the first-degree curve should represent the data. If the data has monotonicity, a turn-back point should not be shown in the figure. Requirements necessary to construct the curve while satisfying these constraints can be summarized to be three properties as follows:

Property 1. Each parametric curve should have a same range as the data.

Property 2. Each parametric curve should appear as increasing curve during the range of data that are increasing and decreasing curve during the range of data decreasing.

Property 3. Turn-back point should appear at the curve only when turn-back point exists in data.

2.5. Characteristics of Curve for the Given Data

To ensure the curve's model more clearly, it is necessary to suggest a critical point.

At a data with the critical value, the curve has maximum and minimum value at neighbor areas close to the given data in case the value is higher or lower than the variable at the neighboring points of two points.

Therefore, it is more reasonable to allow the interpolated curve to prevent from deviating far from the critical value rather than to assign turn-back point with the value identical to the critical value.

When dv_{i+1} and dv_i has opposite sign, the value of data v_i should be recognized to be a critical point.

If 0 is assigned to a function at the given v_i, the interpolated curve has turn-back point. Thus, $S(dv_{i-1}, dv_i)=0$ is required, only when dv_{i-1} and dv_i has opposite sign. If this condition is combined to the drawing illustrated by [Figure 2-2], then domain where S resides will get smaller.

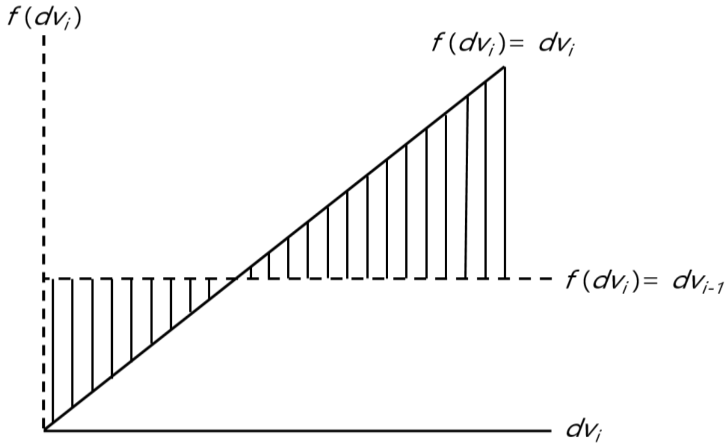


Figure 2-3. Domain of Slope about Critical Point

2.6. Domain of Monotone Curve

Turn-back point should not appear in a set of data of the interpolation equation.

Fritsch and Carlson [4] analyze the interpolated model with three-dimensional curve shape between two values of data. They prove and build this model by combining correlation values of a slope S_i that is used for creating monotone curve as necessary. The result yielded by Fritsch and Carlson is shown in [Figure 2-4].

Therefore, a function S_i should have constraints, when $dv_{i-1} \times dv_i > 0$. Furthermore, a slope S_i cannot exceed three times of the smaller value between dv_i and dv_{i+1}.

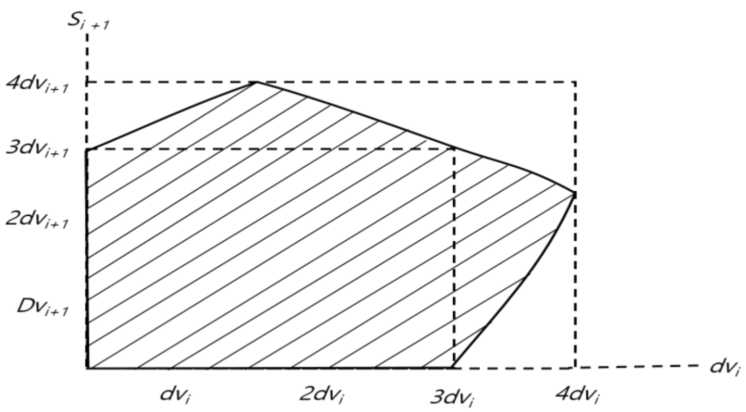


Figure 2-4. The Results of Fritsch and Carlson

3. Slope Algorithm

3.1. Propose Slope Algorithm

As discussed in chapter 2, a slope s_i to calculate the derivatives for a parameter t when values of the given data are provided should derive an interpolated curve within the limited domain as shown in [Figure 2.4].

Following symmetric function for dv_{i-1} and dv_i lies in the limited domain.

If the function is defined as $S=3(dv_{i-1} \times dv_i)/2(dv_{i-1} \times dv_i)$ where $dv_{i-1} \times dv_i$, it appears to be monotonically increasing for $dv_{i-1} \times dv_i$ and symmetric for dv_i and dv_i.

3.2. Validity Analysis

In order to analyze validity of the slope $S=3(dv_{i-1} \times dv_i)/2(dv_{i-1} \times dv_i)$, this Section sequentially describes the procedures as discussed in chapter 2.

First, based on the characteristics of the curve represented by values of the given data in Section 2.3, the slope satisfies a boundary $S_i(dv_i, dv_{i-1}), e(|dv_i|,|dv_{i-1}|)$ within which the curve can be drawn.

Second, dv_i and dv_i are opposite sign at values of the given data, critical value exists. The slope turns to 0 at point i.

Third, based on discussion in Section 2.6, the slope is $S_i(dv_i, dv_{i-1})$, $e(0, min(|3dv_i|, |3dv_{i-1}|))$when $dv_{i-1} \times dv_i > 0$. Thus, the domain meeting all conditions mentioned above can be drawn as [Figure 3-1].

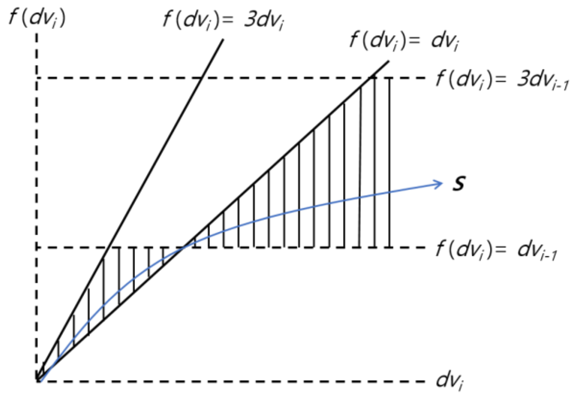


Figure 3-1. Validity Analysis of Slope Assignment Algorithm

4. Smooth Curve Algorithm

4.1. Divide Algorithm

As mentioned earlier, interpolation with equal interval might produce a curve with lack of smoothness. To improve this, we propose an algorithm that enables to construct interpolating interval corresponding to the curvature ratio as follows.

Two separate functions are as follows.

$f(x) = A_0x + A_1xt + A_2xt^2 + A_3xt^3 + \ldots$ (1)

$f(y) = A_0y + A_1yt + A_2yt^2 + A_3yt^3 + \ldots$ (2)

Here, if we find the second order derivatives from two separate functions and then make them 0, we can easily distinguish critical points at which curvature changes.

That is, $t = a_2x / 3a_2x$, $t = a_2y / 3a_2y$ are yielded. When two points are residing in the $0 < t < 1$, they are accepted as an interval. Otherwise, that is, $t \leq 0$ or $t \geq 1$, they are ignored. Therefore, an interval is divided into three sub-intervals if both of two points are accepted within the given interval. If only one of the points is accepted, an interval is divided into two sub-intervals.

If there are inflection points in sub-divided intervals, ratio of an angle of tangent line at the inflection point to an angle of tangent line at the starting point is subdivided with equal interval. Then, interpolation is carried out by subdividing those intervals again.

Thus, subdividing the interval corresponding to the ratio of two angles in the interval can enhance smoothness.

4.2. Smooth Curve Algorithm

Steps to propose the smooth curve algorithm are as follows. Based on a theoretical background on design of an algorithm as discussed in chapter 2, first step is to set the direction for getting smooth curve algorithm. The second step is to apply the slope algorithm suggested in chapter 3 and to define two different functions. The third step is to create the interpolation intervals in conjunction with the divide algorithm as summarized in Section 4.1 The final step is to propose the curve algorithm adopting spline function based on the sub-divided intervals generated in the previous step.

Algorithm 4.1 Smooth Curve Algorithm

```
// Parametric PC method is used for Interpolation
// Draw smooth curve using n points
// x(i),y(i) are the given data set
// I2, I1 are position values of first and second vector

Begin
Initial Nchord, Join, I;
Init=i1;
Determine where closed curve
Draw (x(init),y(init));
Index point increase
If i1>n then return
Else if t1 >n then goto 14
                  Elsel if Nchord=1 then goto 12
                     Else if Nchord<1 then goto 13
Calculate cubic coefficient (ax1, fx, dxi1, ay1, fy, dvi1)
if straight line then straight (xi,xi+1,yi,yi+1,si)
Division (Ax3, Axz, Ay2, Ay3) // Find Any point where curvature Direction
For (i=1,nt,i++){
      Calculate interpolate internal length
                  T=t-dt;
      While ( k<Noiv(i){
      interpolation at (x(i),y(i))
                  K=k+1;
      } ;
} ;
Draw to data points
Calculate curve gradient
2 chord decide on curve at (x(i),y(i)
Calculate gradient at start of open curve
// After last point, decide on curve at (x(i),y(i))
End
```

4.3. Implementation

To implement the smooth curve algorithm, we generated two parametric functions by applying the slope algorithm proposed in chapter 3. Next, we created the interpolation intervals adopting the divide algorithm in Section 4.1. Finally, spline function is applied.

As for boundary condition, the boundary condition of spline function is employed to the slope s_i, which was proposed in the slope algorithm. In this fashion, the way that open curve's boundary is used to assign a slope was implemented by referring to "A Practical guide to Spline" written by De Boor [3].

The slope used here is $S_n = [3dv_{n-1} - s\,(dv_{n-1}, dv_n)/2]$

5. Analysis and Review

5.1. Analysis of Mathematical Error

In this Section, we estimate limits of errors for the proposed smooth curve algorithm, P·C Lagrange, P·C Hermite, and P·C Spline.

In estimating limits of an error for $\|F\text{-}P\|$, P denotes an approximate interpolation polynomial equations and $C'(a,b)$ denotes a space for derivative and continuous real variables with degree 4 at an interval (a, b).

1) In the case of $F \in C'(a,b)$ and $S_3(x)$ B and B-spline interpolation equation, $\|F\text{-}P\| < \frac{\|f\|}{384\ h^4}$, where $h = \frac{(b-a)}{n}$

The above equation 1) is cited from [3] since it has already been proved in [3].

The limits of error for the smooth curve algorithm can be written as follows:

$$\|f\text{-}S\cdot C\| = (t-i)^2(t-i+1)^2[i_0, i, i+1, i+2,]f$$

where $\|f\text{-}S\cdot C\| = (t\text{-}i)^2(t\text{-}i+1)^2[S_{+1} + S_{+2}[f(i)\text{-}f(i+1)]]$

$$\leq\left(\frac{\Delta_i}{2}\right)^2\left(\frac{\Delta_i}{2}\right)^2 \times [f^4(f_x)]\,/\,4i \text{ where, } f_x \in [i, i+1]$$

Thus the limits of error for the smooth curve algorithm are

$$\|f\text{-}S\cdot C\| \leq \frac{\|f^4\|\Delta^4_i}{384}$$ where Δ_i is a size of an interpolation interval

5.2. Numerical Analysis

Numerical analysis is carried out by utilizing functions and numbers proposed by "about piecewise third order interpolating polynomial". As for examples other than this, we compare and investigate data suggested by Fritsch and Carlson [1].

The results obtained from $F(x)$ = cos2π x at a range of [0, 3.5] applying n=9 are as follows.

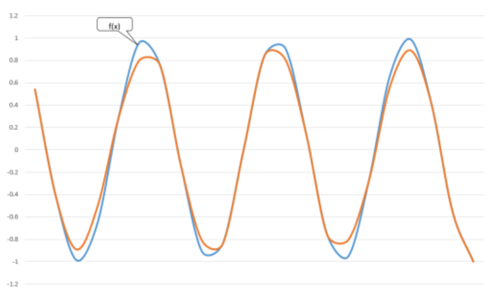


Figure 5-1. $F(x)$ = cos2πx and C. B-Spline

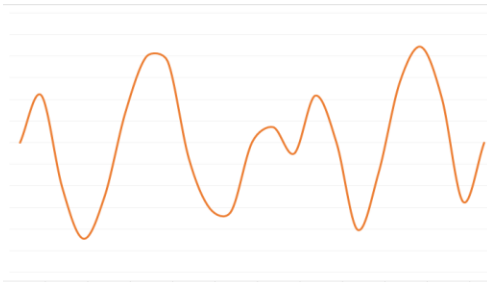


Figure 5-2. Graph of *e*(*x*)=*f*(*x*) - *S*₃(*x*)

The interpolation results of the curve $F(x) = \cos 2\pi x$ at a range of [0, 3.5] applying $n = 9$ by using the smooth curve algorithm are as follows.

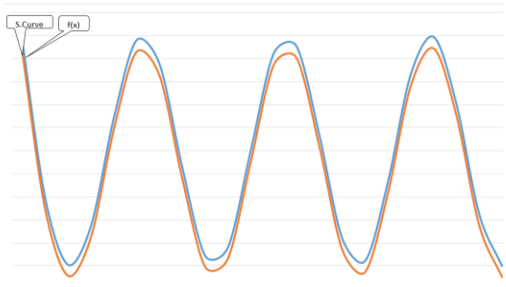


Figure 5-3. $F(x) = \cos 2\pi x$ and C. B-Spline

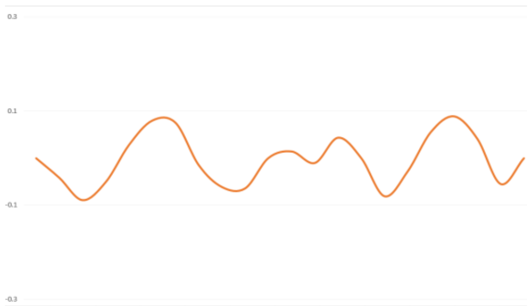


Figure 5-4. Graph of *e*(*x*)=*f*(*x*) - *S·C*(*x*)

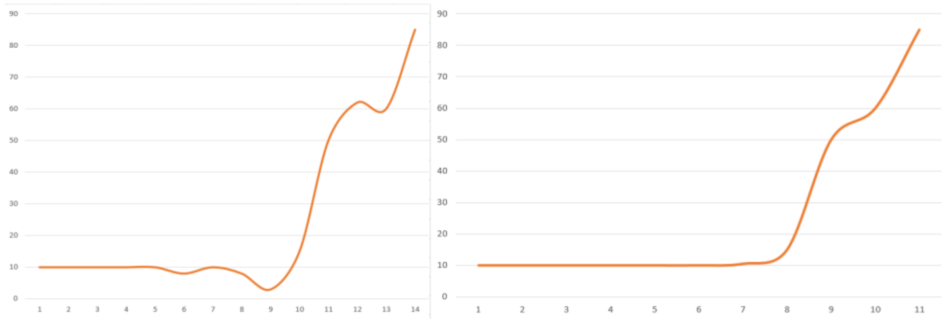
[Table 5.1] compares the results of relative error and absolute error obtained by applying three traditional interpolation methods and the proposed smooth curve algorithm to $F(x) = \cos 2\pi x$ in range of [0, 3.5] at varying $n = 9$, 12 and 15.

Table 5.1 Comparing Error Estimation for $F(x) = \cos 2\pi x$ [0,3.5]

	P·C Lagrange		P·C Hermite		P·C B-spline		P·C smooth curve	
	Absolute	Relative	Absolute	Relative	Absolute	Relative	Absolute	Relative
n=9	0.294	97.72%	0.028	6.39%	0.116	31.34%	0.032	12.28%
n=12	0.117	39.17%	0.009	2.12%	0.023	7.51%	0.015	3.25%
n=15	0.054	19.17%	0.004	0.86%	0.007	1.78%	0007	1.16%

Second, two curves were obtained by applying spline method [4] and the proposed smooth curve algorithm to the data provided by Fritsch and Carlson as following table. [Figure 5.5] illustrates two curves respectively.

x	0	2	3	5	6	8	9	11	12	14	15
y	10	10	10	10	1	01 0	10. 5	15	50	60	85



(a) Mothed: SPLINE (b) Smooth Curve

Figure 5-5. Compare Two Curves for the Data Provided by Fritsch and Carlson

Results obtained by the computer show that the proposed method seems to yield less smoothness than the traditional method. However, this is because we draw the line connecting points after printing the points out. Looking at an actual smooth curve, it is indicated that it is much better than the traditional method in terms of smoothness.

6. Conclusion

This paper proposes an algorithm that assigns proper slope to the data given at two-dimensional space and a divide algorithm that generates interpolating intervals. Based on these algorithms, this paper also proposed an interpolating equation to draw smooth curve.

According to the mathematical error analysis comparing the proposed method to traditional methods, an error rate in P·C interpolation depends on h ,which is an equally divided interval, while an error rate in the proposed method depends on Δ_i yielded by the proposed divide algorithm.

From numerical analysis based on the numerical data, the extent of an error depends on the subdivision ratio of curvature and the position of the given data.

From the results shown in [Table 5.1], the proposed method underperforms the P·C Hermite in terms of precision of an error, while it outperforms the P·C Hermite in terms of accurateness and smoothness.

The proposed algorithm has the advantage that it is easy to apply. The advantage comes from the fact that the monotonicity existing in data is handled by an algorithm, which implies that user or program does not need to care about the monotonicity internally residing in data. Since it is based on only three points, it is said to be compact and fast. Also, it is significantly reasonable because it estimates data on a logical ground.

The algorithm proposed by this paper can be applied to three-dimensional case, which can be surface application. Therefore, it is expected that the proposed method would contribute to various fields like rendering in computer graphics, animation and computer arts, as well as designing and engineering system.

References

[1] K. H. Jine, "On the some piecemise Cubic Interpolation Palynonials", J. of Korean Information Soc., vo1. 10, no. 1, **(1983)**, pp. 55-62.
[2] K. E. Atkinson, "An Introduction to Numerical Analysis", John Wiley & Sons, New Work, **(1978)**, pp.107-153.
[3] C. D. Boor, "A Practical Guide To Splines", Serings-Uerlag, New, York, **(1981)**, pp. 1-57, 314-320.
[4] F. N. Firtsh and R. E. Carlson, "Monoton Piecewise Cubic Interpolation", SIAM. J, Numerical Analysis, vol. 17, **(1980)**, pp. 239-246.
[5] K. Harada and E. Nakmac, "Isotropic Four-point interpolation based on Cubic splines", "Compute Graphics and Image Processing, vol. 28, **(1982)**, pp. 283-287.
[6] J. Stoer and R. Bulirshc, "Introduction to Numerical Analysis", Springer-Usrlang, New York, **(1979)**, pp. 31-147.
[7] G. G. Thomas, "Calculus and Analytic Geometry", **(1957)**, pp. 476-483.
[8] L. W. Johnson and R. Oeanriess, "Numerical Analysis", Addison-Wesley Publishing Company, California, **(1977)**, pp. 187-209.
[9] M. M. Lipschutz, "Differential Geometry", McGraw-Hill Book Company, **(1969)**, pp. 43-79.
[10] O. Herrmann, "Design of non-recursive digital filters with linear phase". Electron Letter, vol. 6, no. 11, **(1970)**, pp.328-329.
[11] Steven C. and Raymond C., "Numerical methods for engineers", McGraw-Hill, **(1998)**, pp. 485-490.
[12] H. S. Nalwa, Editor, "Magnetic Nanostructures", American Scientific Publishers, Los Angeles, **(2003)**.
[13] H. V. Jansen, N. R. Tas and J. W. Berenschot, "Encyclopedia of Nanoscience and Nanotechnology", Edited H. S. Nalwa, American Scientific Publishers, Los Angeles, vol. 5, **(2004)**, pp. 163-275.
[14] J. Kimura and H. Shibasaki, "Recent Advances in Clinical Neurophysiology", Proceedings of the 10th International Congress of EMG and Clinical Neurophysiology, Kyoto, Japan, October 15-19, **(1995)**.

