The International Journal Of Engineering And Science (IJES)
|| Volume || 5 || Issue || 4 || Pages || PP -54-59|| 2016 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Curve Subdivision Approach Available for Embedded System

[1]LUO Xiansong, [2]YANG Li
[1]*Department of Computer Science, Neusoft University of Guangdong,Foshan City, Guangdong Province, PRC*
[2]*Neusoft University of Guangdong,Foshan City, Guangdong Province, PRC*

ABSTRACT
This article presents a subdivision approach that subdivides a parameter curve in a binary style. The new subdivision approach, which is of high precision, is naturally incorporated with a full binary tree and can be realized with a firmware. The approach is valuable in developing embedded system or SoC.
Keywords: *Geometric modeling, Subdivision, Binary tree, Embedded system.*

Date of Submission: 28 March 2016 | Date of Accepted: 11 April 2016

I. INTRODUCTION

Subdivision has been a primary approach in geometric modeling, as surveyed in [1] to [5]. Anyone who has known computer aided geometric design (CAGD) knows that curves and surfaces of CAGD can be generated by De Casteljau or De Boor subdivision, as shown in [6] and [7]. It is also known that these algorithms are only suitable for software of conventional systems. With the development of embedded system, most classical algorithms cannot be afforded to the new system due to limitation of resources in the embedded system, as introduced in [8],[9] and [10]. Therefore, finding new approach that fits the embedded system becomes a new research target for engineers of embedded system. Since 2010, WANG began to publish his researches in treating data of embedded system by binary tree [10]-[13] and applied the approach to fast computation of cubic Bernstein polynomial [14] by firmware. WANG's researches imply that curve subdivision can be performed via direct firmware computation in an embedded system or SoC. This paper extends WANG's approach to a general case.

II. PRELIMINARIES

We need first a simple review of binary tree and subdivision of curves to establish the later contents. Let T be a full binary tree whose depth is n $(n \geq 1)$ and root is T_0 ; let S be an array that stores all the nodes of T in the way from top to bottom and from left to right, and M be an array that stores the sequence of T's inorder traversal. Denote $N_{(k,j)}$ to be T's node at the $j^{th}(j = 0,1,\ldots,2^k - 1)$ position of level $k(k \geq 0)$; then we have the following Lemma 1, which was stated in [10] and [11].

Lemma 1. The index of the node $N(k, j)$ of T is $2^{k-1} + j$ in S , $2^{n-k}(2j+1)$ in M, as shown in Fig.1

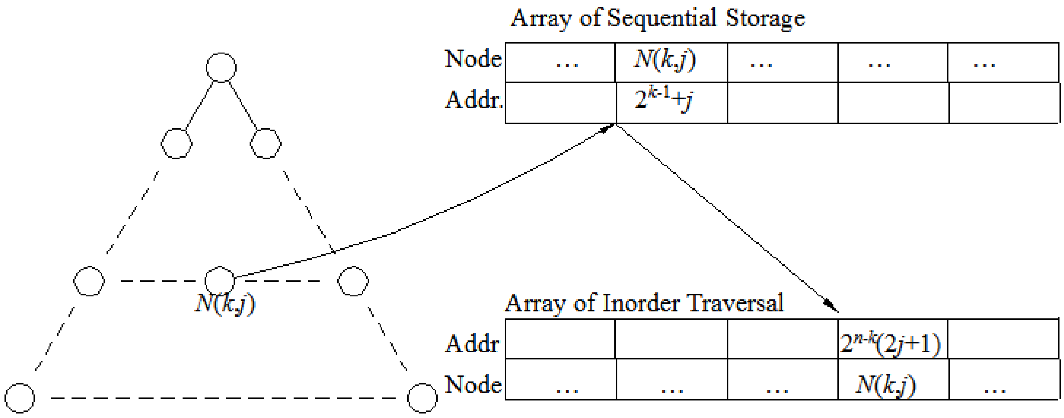


Fig 1. Mapping of T's Node to Its sequential and inorder traversal storage

Lemma 2. Each subdivision of a curve makes the curve into two ones that are of the same property as the original subdivided curve.

Comments: By modeling theory of CG/CAD/CAM, curves are represented by Bezier's model, or B-spline's model or NURBS's model. Each of the models has its own subdivision algorithm. For example, the Bezier curves are subdivided by De Casteljau algorithm, the B-spline curves and the NURBS curves are subdivided by De Boor algorithm. By these algorithms, each subdivision makes the original curve into two sub-curves that are of the same property as the original curve.

III. SUBDIVISION OF BINARY TREE STYLE

In this section we put forward a subdivision that incorporates with a full binary tree and makes an evaluation on the subdivision.

3.1 Curve's Binary Subdivision

Let's first suppose a subdivision is done on an open curve $\beta(t)(t \in \Omega)$, where Ω is a real interval. Without loss of generality, we suppose $\beta(t)$ is regular and so it is homoeomorphous to a line segment and can be reparamerterized into interval $[0,1]$. Figure 2 illustrates its binary subdivision process, in which the circle marks are points from the subdivision.

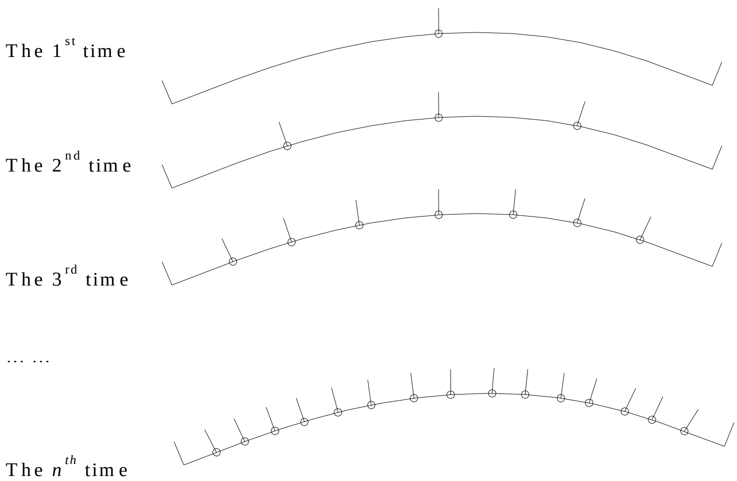


Fig. 2 subdivision process of an open curve

For convenience, we denote the subdivision point from the 1^{st} subdivision by $\beta(\frac{1}{2})$, the two points from the 2^{nd} two subdivisions by $\beta(\frac{1}{4}) = \beta(\frac{1}{2^2})$ and $\beta(\frac{3}{4}) = \beta(\frac{3}{2^2})$, and so on. Then after the n^{th} subdivision, we obtain $2^n - 1$ points and arrange them from the top to bottom and from the left to right by the following sequence

$$\beta(\frac{1}{2}), \beta(\frac{1}{2^2}), \beta(\frac{3}{2^2}), \dots, \beta(\frac{1}{2^n}), \dots, \beta(\frac{2(i-2^{c-1})+1}{2^c}), \dots, \beta(\frac{2^n-1}{2^n}) \tag{1}$$

It is easy to show that the arbitrary item in (1) is to calculated by

$$A_i = \beta(\frac{2(i-2^{c-1})+1}{2^c}), i = 1, 2, \dots, 2^n - 1 \tag{2}$$

where c is the total valid bits in integer i's binary representation, for example, when $i = 1$; $c = 2$ when $i = 2,3$; when $i = 30$, $c = 4$; and in general, $c = k$ when $2^{k-1} \le i \le 2^k - 1$

It is obvious that each item in (1) is one-to-one corresponding to T's nodes if we adopt the following storage strategy

$$\begin{cases} N_{(1,0)} \Leftarrow \beta(\frac{1}{2}) \\ N_{(2,0)} \Leftarrow \beta(\frac{1}{2^2}), N_{(2,1)} \Leftarrow \beta(\frac{3}{2^2}) \\ \quad\quad\quad\quad\quad\quad \cdots\cdots \\ N_{(k,j)} \Leftarrow \beta(\frac{2j+1}{2^k}),\ j = 0,1,\ldots,2^k - 1 \end{cases} \tag{3}$$

In this way, all the subdivision points are stored in a full binary tree, as depicted in figure 3.

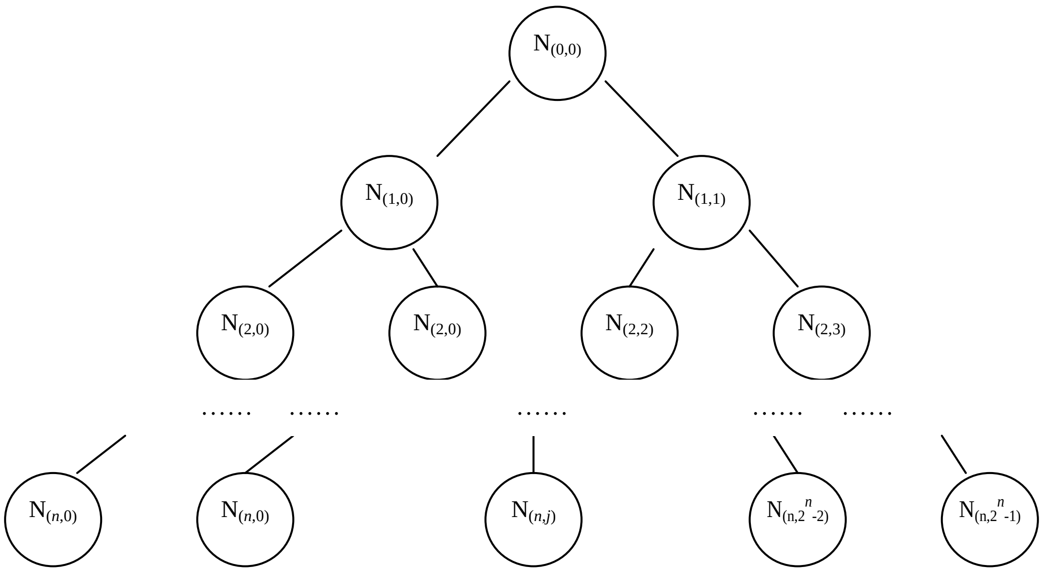


Fig.3 Binary storage of data from subdivision

Let S be a linear storing structure that is used to store the full binary tree; then it knows that S is ***T***'s sequential storage as given bellow.

$$S_0 = \beta(\frac{1}{2}), S_1 = \beta(\frac{1}{2^2}), S_2 = \beta(\frac{3}{2^2}), \ldots, S_{n-1} = \beta(\frac{1}{2^n}), \ldots, S_{2^n-1} = \beta(\frac{2^n - 1}{2^n}) \tag{4}$$

3.2 Properties of the Subdivision

The binary subdivision is proved to have the following properties.
1. Convergence to the curve when $n \to \infty$.

Actually, when $n \to \infty$, $\beta(\frac{1}{2^n}) \to \beta(0), \beta(\frac{2^n - 1}{2^n}) \to \beta(1)$. Hence we know the sequence (1) converges to $\beta(t)(t \in [0,1])$. As $\beta(t)$ is supposed to be regular, it is sure the subdivision converges to the curve.
2. ***T***'s inorder traversal sequence is exactly the sequence of points from the curve's start to its end. Or it exactly and sequentially draws the curve β by taking points from ***T***'s inorder traversal sequence.
In fact, ***T***'s inorder traversal sequence is as follows

$$\beta(\frac{1}{2^n}),\ \beta(\frac{1}{2^{n-1}}),\ \beta(\frac{3}{2^n}), \ldots,\ \beta(\frac{2^{n-1}-1}{2^n}),\ \beta(\frac{1}{2}),\ \beta(\frac{2^{n-1}+1}{2^n})\ , \ldots,\ \beta(\frac{2^n-3}{2^n}),\ \beta(\frac{2^{n-1}-1}{2^{n-1}}),\ \beta(\frac{2^n-1}{2^n}) \tag{5}$$

whose general item is

$$A_i = \beta(\frac{i}{2^n}), i = 1, 2, \ldots, 2^n - 1 \tag{6}$$

It is obvious that (5) is sequentially arranged from the start point to the end point of $\beta(t)$.

3. Approximative precision ε of the subdivision is $\varepsilon = O(\frac{1}{2^n})$.

Actually, by (5) it knows

$$\delta = |\beta(\frac{j+1}{2^n}) - \beta(\frac{j}{2^n})| \approx \frac{1}{2^n} |\beta'(\frac{j}{2^n})|$$

Since $\beta'(t)$ is a continuous real function on $[0,1]$, we know there exists a $L > 0$ such that

$|\beta'(\frac{j}{2^n})| \le L$

That is

$$\delta \le \frac{L}{2^n} \tag{7}$$

On the other hand, since each inter-point, say $\beta(\frac{2j+1}{2^n})$, must have two neighbor points, $\beta(\frac{4j+1}{2^{n+1}})$ and $\beta(\frac{4j+3}{2^{n+1}})$, as shown in figure 4, it yields

$$\Delta = |\beta(\frac{2j+1}{2^n}) - \beta(\frac{4j+1}{2^{n+1}})| = |\beta(\frac{4j+2}{2^{n+1}}) - \beta(\frac{4j+1}{2^{n+1}})| \approx \frac{1}{2^{n+1}} |\beta'(\frac{4j+1}{2^{n+1}})| \le \frac{L}{2^{n+1}}$$

Considering this result and (7), it yields

$$\varepsilon = O(\frac{1}{2^n}) \tag{8}$$

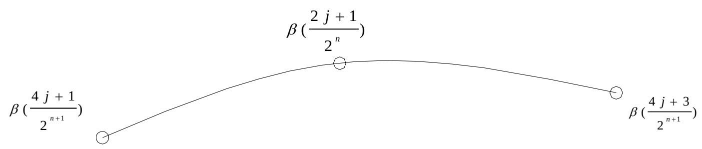
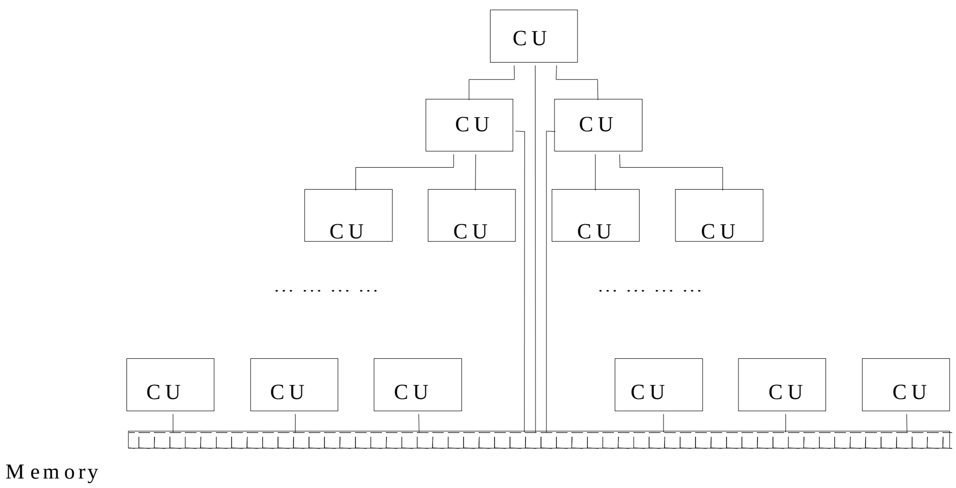


Fig.4 A inter-point and its two neighbors

4. Process of the subdivision can be done by firmware in parallel mode.

For example, we can first design a unit CU to calculate $\beta(\frac{1}{2})$, then the whole sequence (5) can be calculated by an integrated chip that is designed in term of figure 5, in which the unit at the j^{th} position on the k^{th} level, where $k \ge 0$ and $j = 0,1,\ldots,2^k - 1$, outputs its computation to the address that is indexed by $2^{m-k}(2j+1)$ in memory $\boldsymbol{M}$. Obviously, the computation can be performed in parallel mode. Readers may find more details of such design in [16].

Fig.5 Design of parallel computing chip



IV. ALGORITHM TO REGENERATE CURVE

It is seen that our subdivision computes the mid-point $\beta(\frac{1}{2})$. This is unlike the classical algorithms that compute $\beta(\frac{1}{2})$ first. Therefore, it needs a way to obtain $\beta(0)$ and its followers so that a curve can be continuously drawn or shown. Fortunately, we have Lemma 1 that maps an inorder traversal sequence to a sequential sequence. Then we can design algorithm to do the work.

Binary Subdivision Algorithm

Input data:

(1) $\beta(t), \varepsilon$;

(2) Linear storing structure M.

Computing Process

Step 1. Compute depth of binary tree: $n = -\lfloor \log_2 \varepsilon \rfloor$

Step 2. Reallocate length of M to $2^{n+1} - 1$.

Step 3. For k=0 to k=n do

For j=0 to $j = 2^k - 1$ do

$$M[2^{n-k}(2j+1)] = \beta(\frac{2j+1}{2^k})$$

End j

End k

Output data: M that stores β's sequential data.

Note that, for $\varepsilon = 10^{-\alpha}$, we can also estimate n by the following formula

$$n \le \lfloor \alpha(1 + 2.321928) \rfloor \tag{9}$$

V. CONCLUSIONS

Subdivision of a curve of a surface is a primary issue in geometric modeling. Embedded system or SoC requires new approach to perform the their computation. An algorithm that can be realized with a firmware is particularly valuable for an embedded system. This article shows that there exist such algorithms. The author hopes that more valuable approaches will come into being.

REFERENCES

[1] Ben Herbst, Karin M Hunter, Emile Rossouw. Subdivision of Curves and Surfaces: *An Overview, Transactions of the South African Institute of Electrical Engineers*,2007

[2] S Shahid Siddiqi，Kashif Rehan. Curve subdivision schemes for geometric modeling, *International Journal of Computer Mathematics,* 2011, 88(4):851-863

[3] H Kai,Subdivision schemes for geometric modelling: a hands-on tutorial, Dolomites Research Notes on Approximation, *OALIB_Open Access Library*,2012,

[4] M Abdulwahed Abbas. Survey of Subdivision Controls for Interpolating Curves and Surfaces, *International Journal of Software and Informatics,* 2013,7(1):113–132

[5] Pieter J. Barendrecht. IsoGeometric Analysis with Subdivision Surfaces,*11th World Congress on Computational Mechanics (WCCM XI),* 2014, Barcelona, Spain

[6] Jurg Nievercelt, H Klaus. Algorithms and data structures with applications to graphics and geometry, *PRENTICE HALL,* Englewood Cliffs, New Jersey,1993

[7] J F Hughes, Andries van Dam, Morgan McGuire, F D Sklar, D J Foley, K S Feiner, Kurt Akeley,*Computer Graphics: Principles and Practice, (3rd Edition),*Addison-Wesley Professional,2013

[8] Y W Hau，M Khalil-Hani，M N Marsono. CODESL: A Framework for System-Level Modelling, Co-simulation and Design-Space Exploration of Embedded Systems Based on System-on-Chip.*International Conference on Intelligent Systems,Modelling and Simulation*, 2010:122-127

[9] H Tomiyama，T Hieda，N Nishiyama，N Etani. SMYLE OpenCL: A programming framework for embedded many-core SoCs. *Design Automation Conference*, 2013:565-567

[10] LU Schor.*Programming Framework for Reliable and Efficient Embedded Many-Core Systems,ETH-Zürich,* 2014

[11] Wang Xingbo,Storage Strategy and Rapid In-order Traversal Algorithm Available for Reconfigurable Computation of Full Binary Trees, *Journal of Foshan University(Natural Science edition)(In Chinese)*,2010,29(1):47-52

[12] WANG Xing-bo.Fast Algorithms Educed from Intrinsic Properties of Node Indices of Binary Trees, *Computer Engineering and Applications (In Chinese),* 2011，47(9):16-20

[13] Wang Xingbo.Study on Non-recursive and Stack-free Algorithms for Preorder Traversal of Complete Binary Trees, *Computer Engineering and design(In Chinese)*,2011, 32(9):3077-3081

[14] Wang Xingbo. Fast Algorithms for Traversal of Binary Trees Available for SoC, *Computer Engineering and design(In Chinese)*,2013,34(3):874-877
[15] ZENG Yu ,WANG Xingbo. Fast Computation of Cubic Bernstein Polynomials in CNC Interpolation, *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)*,2015,2(1): 29-34
[16] WANG Xingbo. *Method to Design Fast and Rapid CNC Interpolator in Parallel Computation（In Chinese）*, Chinese patent ZL201010593667.6，2014-05-30

AUTHORS' BIOGRAPHY

Luo Xianson was born in Hunan China， He got his Master's degree at Chengdu University of Technology. Since 2002,he has been a teacher of Neusoft Institute,Guangdong. Luo has published 3 books, many papers in computer science.