Science on Form: Proceedings of the First International Symposium for Science on Form,
General Editor: S. Ishizaka, Editors: Y. Kato, R. Takaki, and J. Toriwaki, 281–287.

Recursive Interpolation

Makoto Sato

Department of Computer Science, Tokyo Institute of Technology, Tokyo 152, Japan

Keywords: Interpolation, Recursive Subdivision

We introduce a new method called "recursive interpolation" for interpolation problem. If we use any specific function, it is not easy to provide with simplisity and flexibility. In this method, any specific function form is not used. We use the recursive subdivision scheme to interpolate function values. The algorithm is very simple and easy to calculate. The basis function of the recursive interpolation is investigated and it is shown that interpolated functions and curves are sufficiently smooth and flexible.

INTRODUCTION

Interpolation problem is one of the most basic teqniques not only for signal analysis but also for computer aided design and computer graphics. Many methods have been developed such as Lagrange's interpolation and cubic polynomial spline method. But most of them are computationally expensive and/or quite inflexible.

There have been some works to extend these methods in order to provide with flexibility and versatility. For example, B. Barsky introduced a new spline called Beta spline to provide a means of obtaining local control of bias and tension in piecewise polynomial curves and surfaces (Barsky:1981). But the formulation is not simple enough.

In this paper, we introduce a new interpolation method efficient and flexible. Our basic approaches are as follows:

1. **Procedural Method.** We will not use any specific function such as polynomial, trigonometric, and exponential functions. Because if we use any specific function, it is not easy to provide with flexibility , and even if possible, the formulation becomes complex. We want to define interpolation points directly by procedural modelling.

2. **Recursive Subdivision Method.** We use the recursive subdivision method for interpolation. This method is first introduced by E. Catmull for computer display of curved surfaces (Catmull:1975). After that, A. Fournier used this teqnique to generate fractal curves and surfaces(Fournier:1982). As they pointed out, this method has a lot of advantages for display purposes, that is: (i) the depth of the recursion will be controlled by onscreen resolusion. We never run out of details more than necessary. Therefore, computational efforts are always commensurate with the onscreen image complexity. (ii) The basic computational step in the recursive subdivision uses an interpolation formula. Interpolation formulas are in genaral much easier to compute than incremental ones.

In the next section, the framework of the recursive interpolation method and the interpolation formula are discussed. In the section 3, basic properties of generated curves in this method are investigated using the basis function. The problem to control shapes of interplolated curves is also discussed.

INTERPOLATION ALGORITHM

First we show a mathematical formulation for the scalar-valued interpolation problem. Suppose we are given a uniform knot sequence u_n (= n : integer) and a corresponding set of function values $\{ f_n \}$. The interpolation problem is to find a real function f(u) through these points, i.e.,

$$f(n) = f_n. \tag{1}$$

Here we denote subdivided points of the variable u and corresponding function values as follows:

$$u_n^k := n\cdot\Delta_k \ , \quad \text{where} \quad \Delta_k = 1/2^k \tag{2}$$

and

$$f_n^k := f(u_n^k). \tag{3}$$

In this notation, k indicates the depth of the recursion and n indicates the number ot the points from the origin.

Then the function values f_n^k are defined in the following way:

$$f_n^0 := f_n \qquad \text{for } n = ..,0,1,2,... \tag{4}$$

$$f_n^k := \begin{cases} f_m^{k-1} & \text{for } n = 2m \\ \alpha_k(f_m^{k-1}+f_{m+1}^{k-1}) + \beta_k(f_{m-1}^{k-1}+f_{m+2}^{k-1}) & \text{for } n = 2m+1. \end{cases} \tag{5}$$

Since k goes to infinity, the set $\{ u_n^k \}$ becomes the dense set of the real number and the real function f(u) can be determined.

The fitting problem of open and closed curves in the plane can be formulated in the same manner. A sequence of points

$$P_n = [\ x_n,\ y_n\ ] \ , \quad n = 0,1,...,N \tag{6}$$

are given in the plane. We find a smooth curve P(u) = [x(u), y(u)] through these points. The curve P(u) is represented parametrically, where x(u) and y(u) are real functions of the parameter u. We denote

$$P_n^k := P(u_n^k). \qquad (7)$$

Then we define the curve P(u) in the recursive form:

$$P_n^0 := P_n, \quad n = 0,1,\ldots,N. \qquad (8)$$

$$P_n^k := \begin{cases} P_m^{k-1} & \text{for } n = 2m \\ \alpha_k(P_m^{k-1}+P_{m+1}^{k-1}) + \beta_k(P_{m-1}^{k-1}+P_{m+2}^{k-1}) & \text{for } n = 2m+1. \end{cases} \qquad (9)$$

To define the curve P(u) for $0 \leqq u \leqq N$ by the above formula, the additional points P_{-2}, P_{-1}, P_{N+1}, and P_{N+2} are necessary. These points are correspond to the end conditions in the spline interpolation. In the case of closed curves, we set these points as follows:

$$P_{-1}= P_N,\ P_{-2}= P_{N-1},\ P_{N+1}= P_1, \text{ and } P_{N+2}= P_2 \qquad (10)$$

To determine the coefficients α_k and β_k, we choose the following two functions:

$$\text{(a) } f_1(u) = a_1 + a_2u \qquad (11)$$

and

$$\text{(b) } f_2(u) = b_1\cos Au + b_2\sin Au. \qquad (12)$$

And we require these two functions are generated exactly. These requirements seem to be very natural since these are equivalent that two basic primitives in the plane, a straight line and a circle are required to be generated exactly.

From these requirements, the following two conditions for α_k and β_k are derived:

$$(1)\ \alpha_k + \beta_k = 1/2 \qquad (13)$$

and

$$(2)\ \alpha_k\cos A\Delta_k + \beta_k\cos 3A\Delta_k = 1/2. \qquad (14)$$

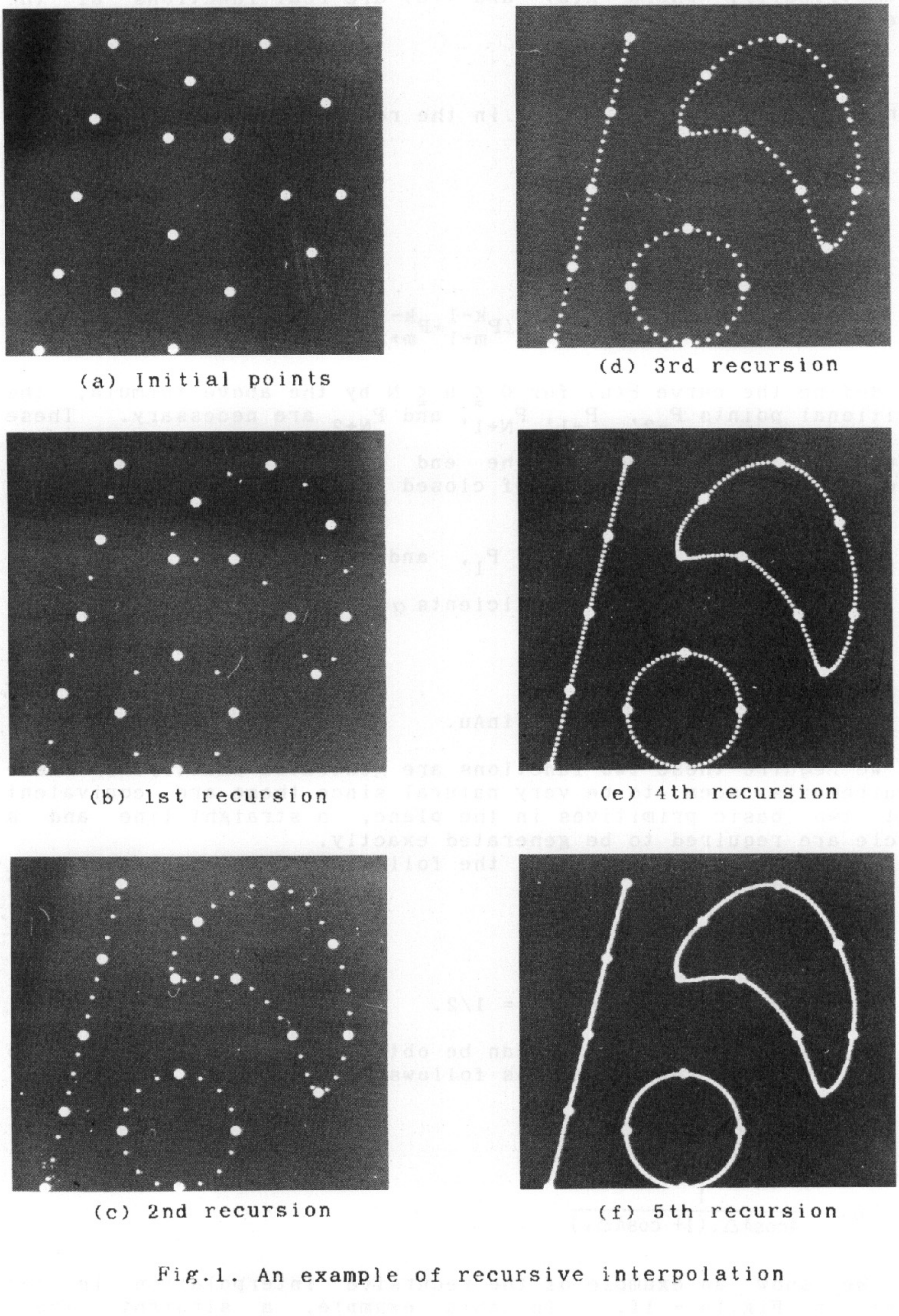
The coefficients α_k and β_k can be obtained by solving these two equations. The results are as follows:

$$\alpha_k=\frac{1}{2}(1+\delta_k)\ ,\ \beta_k=-\frac{1}{2}\delta_k \qquad (15)$$

where

$$\delta_k= \frac{1}{4\cos A\Delta_k(1+ \cos A\Delta_k)}$$

We show an example of the recursive interpolation in the plane in Fig.1a - 1f. In this example, a straight line



(a) Initial points

(d) 3rd recursion

(b) 1st recursion

(e) 4th recursion

(c) 2nd recursion

(f) 5th recursion

Fig.1. An example of recursive interpolation

segment, a small circle, and an arbitrary closed curve are generated.

BASIC PROPERTIES OF RECURSIVE INTERPOLATION

One of the most important properties of functions and curves is the continuity property. Generally speaking, if the 0^{th} through d^{th} derivatives are everywhere continuous (in particular at the knots), then the function or the curve is said to be C^d continuous. For the recursive interpolation, the following continuity property holds.

Theorem 1. Functions and curves generated by the recursive interpolation method are C^1 continuous.

Generally, an interpolation function f(u) is represented in the following form:

$$f(u) = \sum_n f_n \phi(u-n), \tag{16}$$

where the function $\phi(u)$ is called a basis function or a blending function. The shape of the basis function $\phi(u)$ characterizes the properties of the interpolation.

The basis function $\phi(u)$ of the recursive interpolation has a very smooth wave form as shown in Fig.2. This function has the following properties:

(1) $\phi(0) = 1$ and $\phi(n) = 0$ for $n \neq 0$.

(2) $\phi(u) = 0$ for $|u| \geq 3$.

(3) $\phi(3-\Delta_k) = 0$ and $\phi(-3+\Delta_k) = 0$ for $k = 0,1,2,...$

The property (1) shows the basis function $\phi(u)$ being a sampling function. This property garantees the following equations:

$$f(n) = f_n, \quad n = ..,0,1,2,... \tag{17}$$

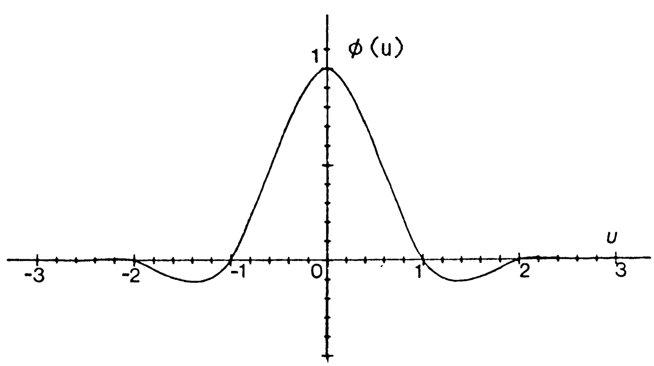


Fig.2. Basis function $\phi(u)$

The property (2) means that the basis function has the local support [-3,3] which shows the local control property of the recursive interpolation. If the function value f_n is varied, the function f(u) varies its function value only in the range [n-3,n+3].

The basis function $\phi(u)$ rapidly coverges to zero as $u \to +3$ and $u \to -3$ as shown in Fig.2. The property (3) shows that the basis function $\phi(u)$ has the infinite zero points near at u = 3 and u = -3, which means that the basis function $\phi(u)$ can not be expressed in terms of usual analytic functions.

Finally, we show that the shape of functions and curves interpolated by this method can be controlled by cnanging the value A of the coefficients α_k and β_k. In Fig.3a-3f, simple examples are shown. From these examples, it is clear that the value A of the coefficients α_k and β_k is related to the tension of plane curves.

CONCLUSION

We proposed a new interpolation method based on the recursive subdivision scheme. In this method, any specific function form is not used, hence the algorithm is very simple and easy to calculate. It is shown that interpolated functions and curves by this method are very smooth and flexible.

REFERENCES

Barsky, B. (1981): The Beta-spline: A local representation based on shape parameters and fundamental geometric measures, Ph. D. dissertation, Depertment of Computer Science, University of Utah.

Catmull, E. (1975): Computer display of curved surfaces, Proc. IEEE Conference on Computer Graphics, Pattern Recognition and Data Structure.

Fournier, A., Fussell, D., and Carpenter, L.(1982): Computer rendering of stochastic models, Comm. of ACM, 25, 6, 371-384.

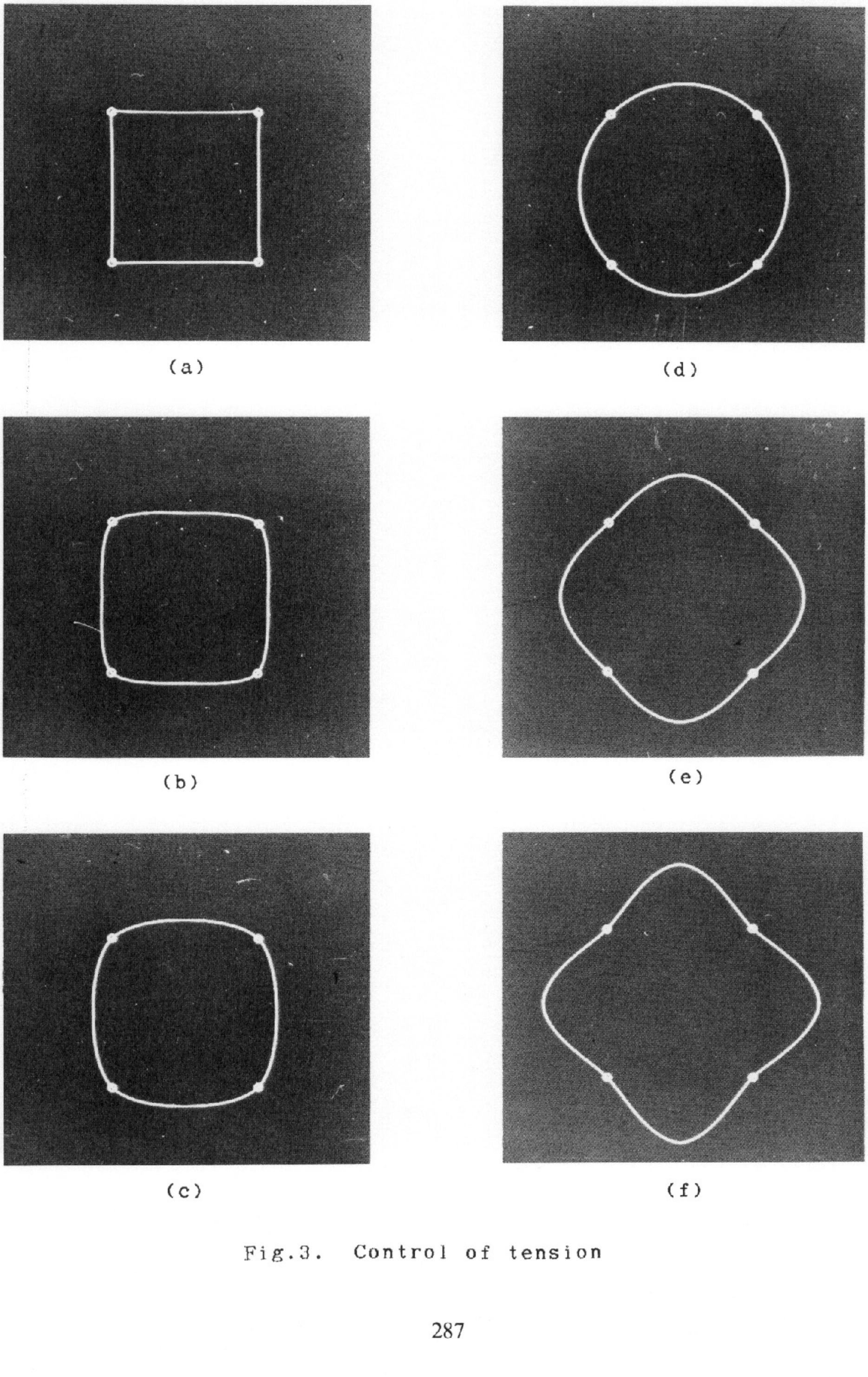


Fig.3. Control of tension