International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:1, No:8, 2007

An Approach to Polynomial Curve Comparison in Geometric Object Database

Chanon Aphirukmatakun, Natasha Dejdumrong

***Abstract*—In image processing and visualization, comparing two bitmapped images needs to be compared from their pixels by matching pixel-by-pixel. Consequently, it takes a lot of computational time while the comparison of two vector-based images is significantly faster. Sometimes these raster graphics images can be approximately converted into the vector-based images by various techniques. After conversion, the problem of comparing two raster graphics images can be reduced to the problem of comparing vector graphics images. Hence, the problem of comparing pixel-by-pixel can be reduced to the problem of polynomial comparisons. In computer aided geometric design (CAGD), the vector graphics images are the composition of curves and surfaces. Curves are defined by a sequence of control points and their polynomials. In this paper, the control points will be considerably used to compare curves. The same curves after relocated or rotated are treated to be equivalent while two curves after different scaled are considered to be similar curves. This paper proposed an algorithm for comparing the polynomial curves by using the control points for equivalence and similarity. In addition, the geometric object-oriented database used to keep the curve information has also been defined in XML format for further used in curve comparisons.

***Keywords*—Bézier curve, Said-Ball curve, Wang-Ball curve, DP curve, CAGD, comparison, geometric object database.

I. Introduction

In Internet applications, there are many popular search engines such as Google, Yahoo and MSN. These search engines have provided efficient mechanisms in searching for the relevant media or documents from a given set of keywords. Nevertheless, textual comparisons can only be accomplished. Although some pictures can be matched to the given keywords, the comparisons are attained by matching from the information provided in those pictures. However, there are several multimedia depicted in the forms of pictures, figures or images. Thus, it is interesting to introduce the algorithms for seeking for these kinds of information.

Typically, images can be classified into two categories: raster graphics and vector graphics images. A raster graphics image is represented by a rectangular grid of pixels whereas vector graphics image is defined by a set of mathematical equations representing the geometric objects, e.g., points, lines, polygons, curves, and surfaces.

The information contained in the raster graphics image is a collection of pixel attributes: the coordinates and colors. Comparing two images, one needs to compare pixel-by-pixel, coordinate-by-coordinate or even color-by-color. Consequently, comparing two large images, e.g., photographic images, will take a lot of time. Searching for a required image in the Internet or in the image banks, it is inevitably needed to compare a plenty of images. Moreover, seeking for a simple geometric object composed in a bitmap image is concerned as a complicated task. In vector-based images, each element is represented in terms of the mathematic formula and its attributes. There are many properties of those geometric primitives that can make the image comparison easier. Using the relevant properties instead of computing the whole image can reduce the computational time. Fortunately in some particular applications, raster graphics images can be converted into the vector-based images. It is reasonable to transform raster graphics images into vector graphics images and compares those vector graphics images.

In computer aided geometric design(CAGD), a vector graphics image is an aggregation of curves and surfaces. Curves and surfaces can be modeled in various techniques. One of those methods that has been commonly used is the polynomial curve and surface representation. There are several kinds of polynomial curves in CAGD, e.g., Bézier, Said-Ball, Wang-Ball, and DP curves. These curves have some common and different properties. All of them are defined in terms of the sum of product of their blending functions and control points. They are just different in their own basis polynomials. In order to compare these curves, we need to consider these properties. The common properties of these curves are control points, weights, and their number of degrees. Control points are the points that affect to the shape of the curve. Weights can be treated as the indicators to control how much each control point influences to the curve. Number of degree determines the maximum degree of polynomials. In different curves, these properties are not computed by the same method. To compare different kinds of curves we need to convert these curves into an intermediate form.

A curve in CAGD can be considered as a geometric object located within a coordinate system. Curves can be relocated. The same curves that are relocated are considered to be the same curves. Curves can be rotated within the coordinate system. Rotated curves can be also treated as the same curves. In some specific cases, curves that are scaled will be considered to be similar curve because they have the same shape but the extents of curves are not the same. Hence, these transformations will be taken into account for the curve comparisons.

In this paper, a framework has been defined for collecting the curve attributes and futher used in curve comparisons. The approach provides an algorithm to compare for curve equivalence, curve similarity and curve inequality. In practise,

Chanon Aphirukmatakun is currently studying at the King Mongkut's University of Technology Thonburi, Bangkok, email: chanon_ap@hotmail.com
Natasha Dejdumrong is working at the King Mongkut's University of Technology Thonburi, Bangkok, email: natasha@cpe.kmutt.ac.th

this technique is faster than the algorithms for comparing bitmapped pictures because it compares only significant properties.

II. Definitions

Bézier curves were firstly developed by de Casteljau in 1959[5] and publicized later by P. Bézier in 1962. In this work, Bézier curves will be used as intermediate curves for comparing two curves because Bézier curves use less computational time and computational space in degree elevation.

Bézier curves of degree *n* with *n*+1 control points, $\{\mathbf{b}_i\}_{i=0}^{n}$ can be defined by:

$$B(t) = \sum_{i=0}^{n} \mathbf{b}_i B_i^n(t), \quad 0 \leq t \leq 1, \tag{1}$$

where $B_i^n(t)$ are Bernstein polynomials defined by:

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}. \tag{2}$$

A. Degree Elevation of Bézier Curves

The new control points of degree *n*+1,$\{\mathbf{b}_i^{(1)}\}_{i=0}^{n+1}$, from an n^{th} Bézier curve can be defined in terms of its control points,$\{\mathbf{b}_i^{(1)}\}_{i=0}^{n}$, by:

$$\mathbf{b}_i^{(1)} = \frac{i}{n+1}\mathbf{b}_{i-1} + (1 - \frac{i}{n+1})\mathbf{b}_i, \tag{3}$$

where *i* = 0,1,2,..,*n*+1.

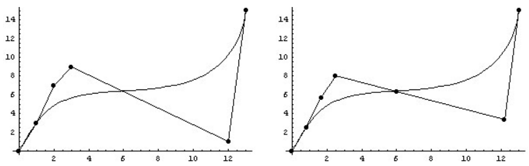

Fig. 1. *Degree Elevation of Bézier curves.*

B. Said-Ball Curves

In 1974, Ball [1][2][3] defined a set of basis functions for cubic curves. In 1989, Said [9] generalized the Ball model to higher degrees and developed the recursive algorithms for two generalized Ball curves. Degree elevation and degree reduction for Said-Ball curves were provided by Hu et al. [6] in 1996.

For *n*+1 control points, denoted by $\{\mathbf{V}_i\}_{i=0}^{n}$, a Said-Ball curve of degree *n* can be expressed by:

$$S(t) = \sum_{i=0}^{n} \mathbf{V}_i S_i^n(t), \quad 0 \leq t \leq 1, \tag{4}$$

where $S_i^n(t)$ are the Said-Ball basis functions defined as following.

$$S_i^n(t) = \begin{cases} \binom{\frac{n-1}{2}+i}{i} t^i (1-t)^{\frac{n+1}{2}} & , \text{ for } 0 \leq i \leq \frac{n-1}{2} \\ \binom{\frac{n-1}{2}+n-i}{n-i} t^{\frac{n+1}{2}} (1-t)^{n-i} & , \text{ for } \frac{n+1}{2} \leq i \leq n \end{cases}, \tag{5}$$

where *n* is odd.

$$S_i^n(t) = \begin{cases} \binom{\frac{n}{2}+i}{i} t^i (1-t)^{\frac{n}{2}+1} & , \text{ for } 0 \leq i \leq \frac{n}{2}-1 \\ \binom{n}{\frac{n}{2}} t^{\frac{n}{2}} (1-t)^{\frac{n}{2}} & , \text{ for } i = \frac{n}{2} \\ \binom{\frac{n}{2}+n-i}{n-i} t^{\frac{n}{2}+1} (1-t)^{n-i} & , \text{ for } \frac{n}{2}+1 \leq i \leq n \end{cases}, \tag{6}$$

where *n* is even.

C. Wang-Ball Curves

Wang-Ball curves were implemented by Wang [11] in 1987 but publicized later in 1996 by Hu et al. [6] with degree elevation and degree reduction.

Wang-Ball curves with *n*+1 control points, denoted by $\{\mathbf{p}_i\}_{i=0}^{n}$ can be formulated by:

$$A(t) = \sum_{i=0}^{n} \mathbf{p}_i A_i^n(t), \quad 0 \leq t \leq 1, \tag{7}$$

where $A_i^n(t)$ are the Wang-Ball polynomials defined as follows:

$$A_i^n(t) = \begin{cases} (1-t)^{2+i}(2t)^i & , \text{ for } 0 \leq i \leq \frac{n-3}{2} \\ (1-t)^{\frac{n+1}{2}}(2t)^{\frac{n-1}{2}} & , \text{ for } i = \frac{n-1}{2} \\ (2(1-t))^{\frac{n-1}{2}} t^{\frac{n+1}{2}} & , \text{ for } i = \frac{n+1}{2} \\ (2t(1-t))^{n-i} t^{n-i+2} & , \text{ for } \frac{n+3}{2} \leq i \leq n \end{cases}, \tag{8}$$

where *n* is odd.

$$A_i^n(t) = \begin{cases} (1-t)^{2+i}(2t)^i & , \text{ for } 0 \leq i \leq \frac{n}{2}-1 \\ (2t(1-t))^{\frac{n}{2}} & , \text{ for } i = \frac{n}{2} \\ (2(1-t))^{n-i} t^{n-i+2} & , \text{ for } \frac{n}{2}+1 \leq i \leq n \end{cases}, \tag{9}$$

where *n* is even.

D. DP Curves

DP curves were developed by Delgado and Penã in 2003[4]. They introduced a new curve representation with linear computational complexity.

DP curve with *n*+1 control points, given by $\{\mathbf{d}_i\}_{i=0}^{n}$, can be expressed by:

$$C(t) = \sum_{i=0}^{n} \mathbf{d}_i C_i^n(t), \quad 0 \leq t \leq 1, \tag{10}$$

where $C_i^n(t)$ are the DP blending functions given by:

$$C_i^n(t) = \begin{cases} (1-t)^n & , \text{ for } i = 0 \\ t(1-t)^{n-i} & , \text{ for } 1 \leq i \leq \lfloor \frac{n}{2} \rfloor - 1 \\ K_1^n(t) + K_2^n(t) & , \text{ for } i = \lfloor \frac{n}{2} \rfloor \\ K_1^n(t) + K_3^n(t) & , \text{ for } i = \lceil \frac{n}{2} \rceil \\ C_{n-i}^n(1-t) & , \text{ for } \lceil \frac{n}{2} \rceil + 1 \leq i \leq n \end{cases}, \tag{11}$$

where

$$K_1^n(t) = \tfrac{1}{2}^{\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor}(1 - t^{\lfloor \frac{n}{2} \rfloor + 1} - (1-t)^{\lfloor \frac{n}{2} \rfloor + 1}),$$

$$K_2^n(t) = (\lceil \tfrac{n}{2} \rceil - \lfloor \tfrac{n}{2} \rfloor) t (1-t)^{\lfloor \frac{n}{2} \rfloor + 1},$$

$$K_3^n(t) = (\lceil \tfrac{n}{2} \rceil - \lfloor \tfrac{n}{2} \rfloor) t^{\lfloor \frac{n}{2} \rfloor + 1}(1-t).$$

E. Conversion Algorithm from Said-Ball into Bézier control points

In 1996, Hu et al.[6] defined algorithms for converting from Said-Ball and Wang-Ball control points into the Bézier control points and vice versa without proof. Later 1997, Tien [10] proved for the conversion by using polar form approach. The Bézier control points, $\{\mathbf{b}_i\}_{i=0}^n$, of Said-Ball curve are determined by:

$$[\mathbf{b}_0 \, \mathbf{b}_1 \, ... \, \mathbf{b}_n] = [\mathbf{V}_0 \, \mathbf{V}_1 \, ... \, \mathbf{V}_n] \begin{bmatrix} b_{00} & b_{01} & ... & b_{0n} \\ b_{10} & b_{11} & ... & b_{1n} \\ ... & ... & ... & ... \\ b_{n0} & b_{n1} & ... & b_{nn} \end{bmatrix}, \tag{12}$$

where $\{\mathbf{b}_i\}_{i=0}^n$ and $\{\mathbf{V}_i\}_{i=0}^n$ are Bézier and Said-Ball control points of a given Said-Ball curve.
The conversion matrix from Said-Ball control point into Bézier control point can be given by:

$$b_{i,j} = \begin{cases} \dfrac{\binom{\frac{n}{2}+i}{i}\binom{\frac{n}{2}-1-i}{j-i}}{\binom{n}{j}} & , \quad \text{for } i \le j \le \frac{n}{2} - 1 \\ \dfrac{\binom{\frac{3}{2}n-i}{n-i}\binom{i-1-\frac{n}{2}}{i-j}}{\binom{n}{j}} & , \quad \text{for } i \ge j \ge \frac{n}{2} + 1 \\ 1 & , \quad \text{for } i = j = \frac{n}{2} \\ 0 & , \quad \text{otherwise} \end{cases}, \tag{13}$$

where *n* is even.

$$b_{i,j} = \begin{cases} \dfrac{\binom{\frac{n-1}{2}+i}{i}\binom{\frac{n-1}{2}-i}{j-i}}{\binom{n}{j}} & , \quad \text{for } i \le j \le \frac{n-1}{2} \\ \dfrac{\binom{\frac{3n-1}{2}-i}{n-i}\binom{i-\frac{n+1}{2}}{i-j}}{\binom{n}{j}} & , \quad \text{for } i \ge j \ge \frac{n+1}{2} \end{cases}, \tag{14}$$

where *n* is odd.

F. Conversion Algorithm from Wang-Ball into Bézier control points

The Bézier control points $\{\mathbf{b}_i\}_{i=0}^n$ of the Wang-Ball curve are determined by:

$$[\mathbf{b}_0 \, \mathbf{b}_1 \, ... \, \mathbf{b}_n] = [\mathbf{p}_0 \, \mathbf{p}_1 \, ... \, \mathbf{p}_n] \begin{bmatrix} a_{00} & a_{01} & ... & a_{0n} \\ a_{10} & a_{11} & ... & a_{1n} \\ ... & ... & ... & ... \\ a_{n0} & a_{n1} & ... & a_{nn} \end{bmatrix}, \tag{15}$$

where $\{\mathbf{b}_i\}_{i=0}^n$ and $\{\mathbf{p}_i\}_{i=0}^n$ are Bézier and Wang-Ball control points of a given Wang-Ball curve.

The conversion matrix from Wang-Ball into Bézier control point of the same curve of degree *n* can be expressed by:

$$a_{i,k} = \begin{cases} 2^i \dfrac{\binom{n-2-2i}{k-i}}{\binom{n}{k}} & , \quad \text{for } i < \lfloor \frac{n}{2} \rfloor \\ 2^{n-i} \dfrac{\binom{2i-2-n}{i-k}}{\binom{n}{k}} & , \quad \text{for } i > \lceil \frac{n}{2} \rceil \\ \dfrac{2^k}{\binom{n}{k}} & , \quad \text{for } i = k = \lfloor \frac{n}{2} \rfloor \\ \dfrac{2^{n-k}}{\binom{n}{k}} & , \quad \text{for } i = k = \lceil \frac{n}{2} \rceil \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{16}$$

G. Conversion Algorithm from DP into Bézier control points

In 2005 the relationships between DP and Bézier of Jiang and Wang [8] were proposed with proof but they are not correct in some degrees. In 2007, Itsariyawanich [7] rewrote and proved the formula for the conversions between DP and Bézier control points, and vice versa, by using the polar form approach. The Bézier control points of a DP curve are determined by:

$$[\mathbf{b}_0 \, \mathbf{b}_1 \, ... \, \mathbf{b}_n] = [\mathbf{d}_0 \, \mathbf{d}_1 \, ... \, \mathbf{d}_n] \begin{bmatrix} c_{00} & c_{01} & ... & c_{0n} \\ c_{10} & c_{11} & ... & c_{1n} \\ ... & ... & ... & ... \\ c_{n0} & c_{n1} & ... & c_{nn} \end{bmatrix}, \tag{17}$$

where $\{\mathbf{b}_i\}_{i=0}^n$ and $\{\mathbf{d}_i\}_{i=0}^n$ are Bézier and DP control points of a given DP curve.
According to the work in [7], the relationship between DP and Bézier control points can be defined in terms of the matrix by:

$$c_{i,j} = \begin{cases} 1 & , \quad \text{for } i = j = 0 \text{ or } i = j = n \\ \dfrac{j\binom{n-j}{n-i}}{n\binom{n-1}{n-i}} & , \quad \text{for } 1 \le j \le i, 1 \le i \le \lfloor \frac{n}{2} \rfloor - 1 \\ \dfrac{(n-j)\binom{j}{i}}{(n-i)\binom{n}{i}} & , \quad \text{for } i \le j \le n-1, \lceil \frac{n}{2} \rceil + 1 \le i \le n-1 \end{cases}, \tag{18}$$

if *n* is even then

$$c_{i,j} = \begin{cases} 1 - \dfrac{\binom{n-j}{\lfloor \frac{n}{2} \rfloor + 1} - \binom{j}{\lfloor \frac{n}{2} \rfloor + 1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}} & , \quad \text{for } 1 \le j \le \lfloor \frac{n}{2} \rfloor \text{ and } i = \lfloor \frac{n}{2} \rfloor \\ 1 - \dfrac{\binom{j}{\lfloor \frac{n}{2} \rfloor + 1} - \binom{n-j}{\lfloor \frac{n}{2} \rfloor + 1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor + 1}} & , \quad \text{for } \lceil \frac{n}{2} \rceil \le j \le n-1 \text{ and } i = \lfloor \frac{n}{2} \rfloor \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{19}$$

if n is odd then

$$c_{i,j} = \begin{cases} \frac{1}{2}\left[1 - \frac{\binom{n-j}{\lfloor \frac{n}{2} \rfloor +1} - \binom{j}{\lfloor \frac{n}{2} \rfloor +1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor +1}}\right] + \frac{2j\binom{n-j}{\lfloor \frac{n}{2} \rfloor +1}}{(n-1)\binom{n}{\lfloor \frac{n}{2} \rfloor +1}}, \\ \text{for } 1 \leq j \leq \lfloor \frac{n}{2} \rfloor \text{ and } i = \lfloor \frac{n}{2} \rfloor \\ \frac{1}{2}\left[1 - \frac{\binom{n-j}{\lfloor \frac{n}{2} \rfloor +1} - \binom{j}{\lfloor \frac{n}{2} \rfloor +1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor +1}}\right] + \frac{(n-j)\binom{j}{\lfloor \frac{n}{2} \rfloor +1}}{n\binom{n-1}{\lfloor \frac{n}{2} \rfloor +1}}, \\ \text{for } 1 \leq j \leq \lfloor \frac{n}{2} \rfloor \text{ and } i = \lceil \frac{n}{2} \rceil \\ \frac{1}{2}\left[1 - \frac{\binom{j}{\lfloor \frac{n}{2} \rfloor +1} - \binom{n-j}{\lfloor \frac{n}{2} \rfloor +1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor +1}}\right] + \frac{j\binom{n-j}{\lfloor \frac{n}{2} \rfloor +1}}{n\binom{n-1}{\lfloor \frac{n}{2} \rfloor +1}}, \\ \text{for } \lceil \frac{n}{2} \rceil \leq j \leq n-1 \text{ and } i = \lfloor \frac{n}{2} \rfloor \\ \frac{1}{2}\left[1 - \frac{\binom{j}{\lfloor \frac{n}{2} \rfloor +1} - \binom{n-j}{\lfloor \frac{n}{2} \rfloor +1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor +1}}\right] + \frac{2(n-j)\binom{j}{\lfloor \frac{n}{2} \rfloor +1}}{(n-1)\binom{n}{\lfloor \frac{n}{2} \rfloor +1}}, \\ \text{for } \lceil \frac{n}{2} \rceil \leq j \leq n-1 \text{ and } i = \lceil \frac{n}{2} \rceil \\ 0, \text{ otherwise.} \end{cases} \tag{20}$$

III. The Equivalence and Similarity of the Curves

In order to compare two curves we need to define the following definitions for the equivalence and similarity of the curves.

Definition 3.1 Let C be a polynomial curve of degree n. $C = \langle P, F, t, n \rangle$, where P be a sequence of control points $\{p_i\}_{i=0}^{n}$, F be a set of blending functions,denoted by $\{f_i^n(t)\}_{i=0}^{n}$, and t be a parametric variable defined by the curve domain, e.g., $t \in [a, b]$ and $a, b \in R$.

Definition 3.2 Two curves are said to be *equivalent* if they are the same curves under translation and rotation.

Figure 2 shows that the same curves after relocating into different positions are considered to be the same curves. Similarly, the same curves after rotated are also treated to be the same, illustrated in Figure 3.

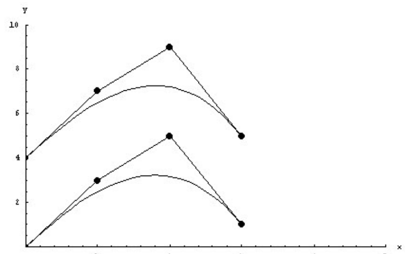

Fig. 2. *The same curves that are translated are said to be the equivalent curves*

Definition 3.3 Let C_1 and C_2 be polynomial curves. C_1 and C_2 are equivalent if $P_{1i} = P_{2i}$, $n_1 = n_2, F_1 = F_2$, $t_1 \subseteq t_2$ and $t_2 \subseteq t_1$ where $i = 0, 1, 2, .., n$

Definition 3.4 Let C_1 and C_2 be polynomial curves. C_1 and C_2 are equivalent if $P_{1i} = P_{2j}$, $n_1 = n_2, F_1 = F_2$, $t_1 \subseteq t_2$

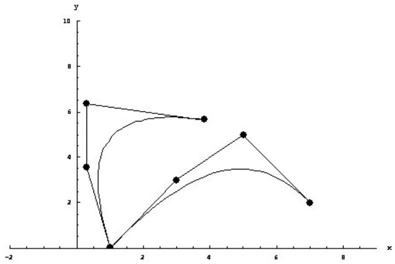

Fig. 3. *The same curves that are rotated are said to be the equivalent curves*

and $t_2 \subseteq t_1$ where $j = n - i$ and $i = 0, 1, 2, .., n$

Definition 3.5 Two curves are said to be *similar* if they are the same curves under scaling.

The same curves that are scaled are considered to be similar curves because their angles of inner control points are equal but their lengths between the control points are different (Figure 4).

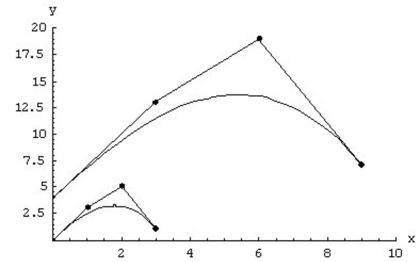

Fig. 4. *The same curves that are scaled are considered to be similar curves*

IV. Algorithm for Comparing the Equivalence of Curves

In this section, we proposed an algorithm for comparing between two curves. The algorithm is used to compare whether two curves are equivalent, similar or different curves. The algorithm uses Bézier curves as an intermediate form because the Bézier curves use less computational time and computational space in computing the degree elevation and they are the best curve approximation from their control points. The steps of comparing two curves can be enumerated as follows:

Algorithm

1. Convert the given curves into the Bézier curves.
2. Apply degree elevation of Bézier curves to raise the different degree of two curves to be the same degree.
3. Measure the lengths between the pair of two consecutive control points, denoted by $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$. Let $\mathbf{p}_i = (x_i, y_i, z_i), \mathbf{p}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, and L_i be lengths between the control points.

$$L_i = \overline{\mathbf{p}_i\mathbf{p}_{i+1}} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}.$$

4. Measure the ratios of lengths between the control points.

International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:1, No:8, 2007

Let R_i be the ratios of lengths between the control points.

$$R_i = \frac{L_i}{\sum_{i=0}^{n} L_i}.$$

5. Measure the angles of inner control points. Let α_i be an angle between each inner control points, U_i be vector from P_{i-1} and P_i, and V_i be vector from P_i and P_{i+1}.

$$U_i.V_i = |U|\,|V| \cos(\alpha_i).$$

$$\alpha_i = \cos^{-1}(\frac{U_i.V_i}{|U|\,|V|}).$$

6. Measure the angles of normal vector between each inner control points. Let β_i be angles of each normal vector, U_i be vector from $\mathbf{p}_{i-1}$ and $\mathbf{p}_i$, V_i be vector from $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$,and N_i be normal vector between U_i and V_i.

$$N_i = U_i \times V_i.$$

$$\beta_i = \cos^{-1}(\frac{N_i.N_{i+1}}{|N_i|\,|N_{i+1}|}).$$

7. Reverse the sequence order of the control points and repeat the steps from 3 to 5 by setting $\mathbf{p}_i = \mathbf{p}_{n-i}$.
8. If all measurements in 3, 4, 5, 6 are equal then two curves are said to be *equivalent*. If only measurement of lenghts between the control points in step 3 is not equal then two curves are *similar*, otherwise, two curves are different.

V. XML DESCRIPTION FOR CRUVES INFORMATION

In order to search for curves, the XML description for collecting the curve information is defined.

A. DTD Modeling

XML Document Type Definitions(DTDs) is used to define an XML document structure. It defines the document structure by the list of attributes and elements. Hence, XML DTD is defined in order to keep the curve information as follows:

Let DTD=$\{d_0, d_1, .., d_n\}$

d_0 : < !ELEMENT Non-Rational_Curves (Curve+)>
d_1 : < !ELEMENT Curve (Blending_Function, Sequence_of_Control_Points, Sequence_of_Lengths, Sequence_of_Length_Ratios, Sequence_of_Inner_Angles, Sequence_of_Angles_of_Normal_Vector, Number_Of_Degree)>
d_2 : <!ATTLIST Curve Curve_ID NMTOKEN #IMPLIED Curve_Name CDATA #IMPLIED >
d_3 : <!ELEMENT Blending_Function (#PCDATA)>
d_4 : <!ELEMENT Sequence_of_Control_Points (Control_Point+)>
d_5 : <!ELEMENT Control_Point (Px, Py, Pz)>
d_6 : <!ATTLIST Control_Point ID NMTOKEN #IMPLIED>
d_7 : <!ELEMENT Px (#PCDATA)>
d_8 : <!ELEMENT Py (#PCDATA)>
d_9 : <!ELEMENT Pz (#PCDATA)>
d_{10} : <!ELEMENT Sequence_of_Lengths (Length+)>
d_{11} : <!ELEMENT Length (Value)>
d_{12} : <!ATTLIST Length ID NMTOKEN #IMPLIED>
d_{13} : <!ELEMENT Value (#PCDATA)>
d_{14} : <!ELEMENT Sequence_of_Length_Ratios (Length_Ratios+)>
d_{15} : <!ELEMENT Length_Ratios (Value)>
d_{16} : <!ATTLIST Length_Ratios ID NMTOKEN #IMPLIED>
d_{17} : <!ELEMENT Sequence_of_Inner_Angles (Inner_Angle+)>
d_{18} : <!ELEMENT Inner_Angle (Value)>
d_{19} : <!ATTLIST Inner_Angle ID NMTOKEN #IMPLIED>
d_{20} : <!ELEMENT Sequence_of_Angles_of_Normal_Vector (Angle_of_Normal_Vector+)>
d_{21} : <!ELEMENT Angle_of_Normal_Vector (Value)>
d_{22} : <!ATTLIST Angle_of_Normal_Vector ID NMTOKEN #IMPLIED>
d_{23} : <!ELEMENT Number_Of_Degree (#PCDATA)>

B. An Example of XML representation for geometric object-oriented database

This is an example of XML database for a curve. It contains the neccessary information for comparing and searching the curve.

Example

```
<Non-Rational_Curves>
 <Curve Curve_ID="0" Curve_Name="Cubic_Curve">
  <Blending_Function>Bézier</Blending_Function>
  <Sequence_of_Control_Points>
   <Control_Point ID="0">
    <Px>0</Px>
    <Py>0</Py>
    <Pz>0</Pz>
   </Control_Point>
   <Control_Point ID="1">
    <Px>5</Px>
    <Py>5</Py>
    <Pz>2</Pz>
   </Control_Point>
   <Control_Point ID="2">
    <Px>5</Px>
    <Py>0</Py>
    <Pz>0</Pz>
   <Control_Point ID="3">
    <Px>10</Px>
    <Py>5</Py>
    <Pz>4</Pz>
   </Control_Point>
  </Sequence_of_Control_Points>
  <Sequence_of_Lengths>
   <Length ID="0">
    <Value>7.34847</Value>
   </Length>
   <Length ID="1">
    <Value>5.38516</Value>
   </Length>
   <Length ID="2">
    <Value>8.12404</Value>
   </Length>
  </Sequence_of_Lengths>
  <Sequence_of_Length_Ratios>
   <Length_Ratios ID="0">
    <Value>0.577091</Value>
```

International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:1, No:8, 2007


```
  </Length_Ratios>
  <Length_Ratios ID="1">
   <Value>0.422909</Value>
  </Length_Ratios>
  <Length_Ratios ID="2">
   <Value>0.389499</Value>
  </Length_Ratios>
 </Sequence_of_Length_Ratios>
 <Sequence_of_Inner_Angles>
  <Inner_Angle ID="0">
   <Value>0.748327</Value>
  </Inner_Angle>
  <Inner_Angle ID="1">
   <Value>0.716212</Value>
  </Inner_Angle>
 </Sequence_of_Inner_Angles>
 <Sequence_of_Angles_of_Normal_Vector>
  <Angle_of_Normal_Vector ID="0">
   <Value>2.78599</Value>
  </Angle_of_Normal_Vector>
 </Sequence_of_Angles_of_Normal_Vector>
 <Number_Of_Degree>3</Number_Of_Degree>
</Curve>
</Non-Rational_Curves>
```


VI. Experiments and Results

A. Comparing the equivalent of the different curves with different degrees.

Given the control points of a cubic Said-Ball curve {(0, 0, 0), (1.5, 0, 4.5), (1.5, 0, 7), (3, 0, 1)} and the control points of a quartic Wang-Ball curve {(0, 1, 0), (0, 2.5, 4.5), (0, 2.5, 5.75),(0, 2.5, 7), (0, 4, 1)}.

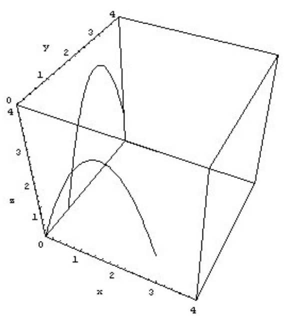

Fig. 5. *The same curves are in different locations. Above curve is Said-Ball curve degree 3 and below curve is Wang-Ball curve degree 4.*

To compare the Said-Ball curve of degree 3 and Wang-Ball curve of degree 4, one needs to convert their control points into the Bézier control points of the same degree. The Bézier control points of Said-Ball curve computed from the conversion matrix (12) are readily obtained by {(0, 0, 0), (0, 0, 3), (2, 0, 5), (3, 0, 1)} and the Bézier control points of Wang-Ball curve getting from the conversion matrix (15) are {(0, 1, 0), (0, 1.75, 2.25), (0, 2.5, 4), (0, 3.25, 4), (0, 4, 1)}. Because the Bézier control points converted from Wang-Ball and Said-Ball control points are not expressed in the same degree. The Bézier control points of Wang-Ball curves have degree 4 that is the maximum degree. Thus, we use the degree elevation (3) to raise the degree of Bézier control points of Said-Ball into degree 4. The Bézier controls points of Said-Ball degree 4 are:{(0, 0, 0), (0.75, 0, 2.25), (1.5, 0, 4), (2.25, 0, 4), (3, 0, 1)}.

The Said-Ball and the Wang-Ball curves can be compared by using the proposed algorithm. The lengths between the control points of Said-Ball curve are (2.372, 1.904, 0.7500, 3.092) and the lengths between the control points of Wang-Ball curve are (2.372, 1.904, 0.7500, 3.092). The ratios of lengths between the control points of Said-Ball curve are (0.292155, 0.234534, 0.0923875, 0.380923) and the ratios of lengths between the control points of Wang-Ball curve are (0.292155, 0.234534, 0.0923875, 0.380923). The inner angles between the control points of Said-Ball curve are(3.058, 1.976, 1.816) and the inner angles between the control points of Wang-Ball curve are (3.058, 1.976, 1.816). The inner angles of normal vectors between the control points of Said-Ball curve are (0, 0) and the inner angles of normal vectors between the control points of Wang-Ball curve are (0, 0). Because the measurments between the control points of the Said-Ball and the Wang-Ball curves are all equal, these Said-Ball and the Wang-Ball curves are said to be *equivalent*.

B. Comparing the similarity of the different curves with different degrees.

Given the control points of a cubic DP curve {(4, 1, 0), (4.5, 0, 0.5), (0., 3, 3.5), (4, 2.5, 0.5)} and the control points of a quartic Bézier curve {(0, 0, 0), (0, 1.5, 2.25), (1, 3.5, 4), (2.25, 3.75, 4), (3, 0, 1)}.

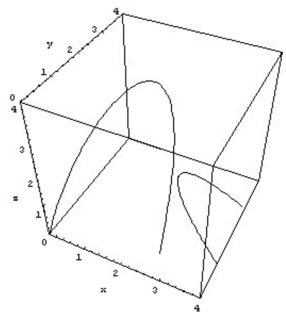

Fig. 6. *The same curves are in different locations. Above curve is DP curve degree 3 and below curve is Bézier curve degree 4.*

To compare DP curve of degree 3 and Bézier curve of degree 4, one needs to convert DP to Bézier curve by using the conversion matrix (17).The Bézier control points of DP curve are {(4, 1, 0), (3, 1, 1.5), (1.5, 2, 2.5), (4., 2.5, 0.5)}. Because the Bézier control points of DP curve and the Bézier control points of Bézier curve are not expressed in the same degree, the degree elevation (3) can be used to raise the degree of the Bézier controls points of DP curve into degree 4. The Bézier controls points of DP curve degree 4 are {(4, 1, 0), (3.25, 1, 1.125), (2.25, 1.5, 2), (2.125, 2.125, 2), (4, 2.5, 0.5)}.

DP and Bézier curves can be compared by using the proposed algorithm. The lengths between the control points of DP curve are (1.352, 1.420, 0.6374, 2.430) and the lengths between the control points of Bézier curve are(2.704, 2.839, 1.275, 4.861). The ratios of lengths between the control points of DP curve are (0.231542, 0.243126, 0.10915, 0.416182) and the ratios of lengths between the control points of Bézier curve are (0.231542, 0.243126, 0.10915, 0.416182). The inner angles between the control points of DP curve are(2.699, 2.075, 1.571) and the inner angles between the control points of Bézier curve are (2.699, 2.075, 1.571). The inner angles of normal vectors between the control points of DP curve are (0.5065, 0.1159) and the inner angles of normal vectors between the control points of Bézier curve are (0.5065, 0.1159). Because only the measurment of lengths between the control points of the DP and the Bézier curves are not equal, the DP and the Bézier curves are not *equivalent* but they are *similar*.

VII. Conclusions

According to the algorithm, it has been found that comparing two curves by using their control points can reduce the number of comparisons because the problem was reduced from comparing pixel-by-pixel into the problem of control point comparisons. The proposed algorithm can compare two curves either represented in different coordinates or after geometric transformations, e.g., translation, rotation, mirror reflection about an axis, scaling, or their combination. In addition, using the XML for keeping the curve information can be used as an intermediate format in searching for the curves. In the future, there may be several formats for vector graphics images. Those formats can be readily converted into XML files. In the Internet applications, the search engine can add on this ability to search for the vector based images. This work will be applied in searching for the vector graphics images in the web documents.

Acknowledgment

This work was supported by CAGD and Animation laboratory (CADLAB), the department of computer engineering, KMUTT.

References

[1] Ball A. A.,1974, *CONSURF, Part one: Introduction to conic lifting title. Computer-Aided Design 6*, pp. 243-249.
[2] Ball A. A.,1975, *CONSURF, Part two: Description of the algorithms. Computer-Aided Design 7*, pp. 237-242.
[3] Ball A. A.,1977, *CONSURF, Part three: How the program is used. Computer-Aided Design 9*,pp. 912.
[4] Delgado, J. and Penã, J.M.,2003,*A Shape Preserving Representation with an evaluation algorithm of linear complexity*, Computer Aided Geometric Design, pp. 1-10.
[5] Farin, G. ,1990, *Curves and Surfaces for Computer Aided Geometric Design* , A Practical Guide, 5nd Edition, Academic Press Inc., London.
[6] Hu, S.M., Wang, G.Z. and Jin, T.G., 1996, *Properties of Two Types of Generalized Ball Curves*, Computer-Aided Design, Vol. 28, No. 2, pp. 125-133.
[7] Itsariyawanich,K. ,2007,*DP curve and its properties*, Master Thesis, King Mongkut's University of Technology Thonburi, Thailand.
[8] Jiang, S.R. and Wang, G.J.,2005,*Conversion and Evaluation for Two Types of Parametric Surface Constructed by NTP Bases*,Computers and Mathematics with Applications, Vol. 49, No. 2, pp. 321-329.
[9] Said, H.B., 1989, *A Generalized Ball Curve and Its Recursive Algorithm*, ACM Transactions on Graphics, Vol. 8, No. 4, pp. 360-371.
[10] Tien, H.L. ,1997,*Further Developments on Ball Curves and Surfaces*, Ph.D Dissertation, Asian Institute of Technology School of Advanced Technologies, Bangkok, Thailand.
[11] Wang, G.Z., 1987, *Ball Curve of High Degree and Its Geometric Propertied*, Appl. Math: A journal of Chinese Universities, Vol. 2, No. 1, pp. 126-140.

Chanon Aphirukmatakun was graduated from the computer engineering department, King Mongkut's University of Technology Thonburi. Nowadays, the author is a graduate student at the department of computer engineering, KMUTT and he is also working as a research assistant at the CADLAB.

Natasha Dejdumrong hold the bachelor degree in computer science from Thammasat university and the master and doctoral degrees from the department of computer science and information management, Asian Institute of Technology. The fields of interest are computer aided geometric design (CAGD), Animation and visual effects. Currently, the author is working at the department of computer engineering, King Mongkut's University of Technology Thonburi and is the chief of the CADLAB.