NURBS Parameterization: A New Method of Parameterization Using the Correlation Relationship between Nodes

Sawssen Jalel, Mohamed Naouai, Atef Hamouda, and Malek Jebabli

Faculty of Science of Tunis, University campus el Manar DSI 2092 Tunis
Belvédaire-Tunisia Laboratory LIPAH
{sawssen.jalel,jebabli.malek}@gmail.com, naouai@polytech.unice.fr,
atef_hammouda@yahoo.fr

Abstract. NURBS (Non-uniform rational B-splines) has become the industry standard tools for the representation, design and data exchange of geometric information to be processed and used by computers because of their useful geometrical properties. The problem of the parameterization of data points in NURBS curve/surface has been considered by several of researchers. We propose in this paper a new parameterization method for NURBS approximation. The current methods of parameterization such as centripetal method uses only the previous knot vector to calculate the recent knot. In this paper, we give a new parameterization method based on the correlation of the nodes. This approach inherits the advantages of the relation and position of the knots.

Keywords: NURBS curves, Approximation, parameterization, knot vector.

1 Introduction

In several applications for describing curves, the two major problems are approximation and interpolation. Among these applications, the field of pattern recognition appear to be strong. Based on the role of curves and surfaces, we focus here on the curves. There are many curves that have been developed so far such as Conic, Hermite, Cubic spline, Nu spline, Beta spline,Bezier, Bspline, NURBS, etc. NURBS curve are widely used in the area of computer aided geometric design (CAGD) and manufacturing due to its simplicity and advantages over the other curves [5,9]. Non-uniform rational B-splines (NURBS) have various useful properties such as smoothness and the possibility of local modifications, which facilitates the representation of general free form surfaces [9]. It is widely used in many research areas such as NURBS Skeleton [3]. These properties of NURBS are ideal for the design of complicated geometry [1]. NURBS has become the industry standard tools for the representation, design and data exchange of geometric information that is to be processed and used by computers because of their useful geometrical properties [4]. NURBS are built from B-splines. The

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2012, LNCS 7329, pp. 216–225, 2012.

B-spline parametric space is local to patches rather than elements. A knot vector in one dimension is a set of coordinates in the parametric space, written $U = \{u_1, u_2, ..., u_{n+p+1}\}$, where $u_i \in \mathbb{R}$ is the ith it knot, i is the knot index, $i = 1, 2, ..., n+p+1$, p is the polynomial order, and n is the number of basis functions which comprise the B-spline[4]. Therefore, the choice of the parameter values affects the shape of the expected resulting NURBS curve/surface. Although many researchers have carried out considerable work on the parameterization problem, there is still a place to improve the parameterization process for interpolating and approximating complicated surfaces in manufacturing industries [4,1,8]. In this paper, a new parameterization method is proposed for NURBS curves interpolation and approximation based on the relation and position of the knot. The rest of this paper is organized as follows. Section 2 introduces NURBS curve formulations, the interpolation and approximation methods which we are going to deal with. In section 3, the proposed parameterization method is defined. In section 4, the output of applying the method is presented and discussed. We conclude in Section 5.

2 NURBS Curves

We recall that a NURBS curve of degree p is defined by [9]:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \ . \tag{1}$$

Where P_i are control points, w_i are the weights associated with these points. Indeed, each w_i determines the influence of point P_i on the curve. $N_{i,p}(u)$ are the B-spline basis functions of degree p defined [9] on the non-periodic and non-uniform knot vector U, recursively by:

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i-1,p-1}(u) \ . \tag{2}$$

Where

$$N_{i,0}(u) = \begin{cases} 1 & \text{if} \quad u_i \leqslant u < u_i + 1 \\ 0 & \text{else} \end{cases} \tag{3}$$

As a consequence, the basis functions become,

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \ . \tag{4}$$

and

$$C(u) = \sum_{i=0}^{n} R_{i,p}(u) P_i \ . \tag{5}$$

$R_{i,p}(u)$ is called NURBS basis function and share some properties of B-Spline and *Bezier* basis function [7].

2.1 Knot Vector

Knot vector $U = \{u_0, u_1, ..., u_{n+p+1}\}$ is a non-decreasing breakpoints sequence of real number [10,8]. u_i are called knot and they satisfy $u_i < u_{i+1},\ i = 0, ..., n+p+1$ .There are two types of knot vector [10], open and periodic knot vector [12]. In a *uniform* knot vector, individual knot values are evenly spaced. Examples are $[0.1\ 0.2\ 0.3\ 0.4\ 0.5], [-0.2\ -0.1\ 0\ 0.1\ 0.2]$. In practice, *uniform* knot vectors generally begin at zero and are incremented by 1 to some maximum value, or are normalized in the range between 0 and 1, i.e., equal decimal intervals; for example [0 0.25 0.5 0.75 1]. An *open uniform* knot vector has multiplicity of knot values at the end equal to the order k of the B-spline basis function. Internal knot values are evenly spaced. Some examples (for $k = 3$) using integer incrementation are [0 0 0 1 2 3 3 3] or for normalized incrementation $[0\ 0\ 0\ \frac{1}{3}\ \frac{2}{3}\ 1\ 1\ 1]$. In this study, open type will be used since it makes the curve clamped on the both end of control points. $U = \left\{ \underbrace{u_0, ..., u_p}_{0}, u_{p+1}, ..., u_n, \underbrace{u_{n+1}, ..., u_{n+p+1}}_{1} \right\}$. The $\{u_{p+1}, ..., u_n\}$ is called internal knot span [1,4,13]. Our proposition concerns the parameterization of this internal knot.

The calculation of the knot vector is through two steps that are the parameterization and generation. We will start this section by presenting the methods of generation.

2.2 Methods for Generating Knot Vector

Equally Spaced. Equally spaced is a method for generating knot vector. It doesn't require the parameter values to generate the knot. The internal knot vector is calculated as below [8,9]:

$$u_{j+p} = \frac{j}{n+p+1} \qquad \text{j=1,...,n-p} \tag{6}$$

However, this method is not recommended by some researchers because it can produce a singular system of equation if it is combined with chord length and centripetal parameterization [9].

Averaging Knot Vector. Another popular method for generating knot vector, suggested by de Boor, is to average the parameters. The internal knot span is generated [8,9,12] as below:

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} u_i, \qquad \text{j=1,...,n-p} \tag{7}$$

By using this method the knots values will distribute along the parameter value and leads to a system of equation which is totally positive [9,12].

Piegl Approximation Knot Vector. This method is used for approximation method only. The internal knot span is calculated as follows [8]:

$$u_{j+p} = (1 - (ja - i))\, u_{i+1} + (ja - i)\, u_i \tag{8}$$

where i=$\lfloor ja \rfloor$, j=1,...,n-p and a=$\frac{m+1}{n-p+1}$

where m+1 is number of data points, n+1 is number of control points, and p is degree of the curve. Equation 8 guarantees that every knot span contains at least one parameter value and leads to positive definite [8].

2.3 Parameterization

The quality of the approximating curve C(u) depends on the selection of parametric values u_i for the given points P_i. At this step, the parameterization describes the distribution of each point on NURBS curve.

Uniform Parameterization Method. The distance between the data points is defined as below:

$$d_i = i \quad \text{i=1,...,m} \tag{9}$$

where **d** is the distance between data points D with number of data points **m+1** [9]. In many cases, interpolation between edit points is not as good. This can lead to unpredictable stretching of textures during rendering and may cause unpleasant results like big bulges, sharp peaks and loops [8,13].

Chord Length Parameterization Method. The distance between the data points is defined as below:

$$d_i = |D_i - D_{i-1}|, \quad \text{i=1,...,m} \tag{10}$$

where **d** is the distance between data points D with number **m+1**[9]. The chord length method is widely used and usually performs well but a longer chord may cause its curve segment to have a bulge bigger than necessary[8,13].

Centripetal Parameterization Method. E. T. Y. Lee proposed this centripetal method. The distance between the data in centripetal method is defined by the following equation:

$$d_i = \sqrt{|D_i - D_{i-1}|}, \quad \text{i=1,...,m} \tag{11}$$

where **d** is the distance between data points D with number **m+1** [9].

Similarly, the centripetal method should work similar to the chord length method because the former is an extension to the latter. However, it is not always the case when the distribution of chord lengths change wildly [8,13]. Commonly, the parameter value is in the range [0, 1] and it depends on the minimum and maximum values of knot vector. The rest of parameter is calculated as follows: $t_0 = 0, t_n = 1$ and

$$t_i = t_{i-1} + \frac{d_i}{\sum_{j=0}^{n-1} d_j}, \quad \text{i=1,...,n} \tag{12}$$

For further reading about parameterization, please refer to [6,9,15].

3 Proposed Approach

We seek a new parameterization of the knot vector. First, we note that the existing parameterizations is dependent on the topology. So we propose an independent parameterization of the marker. To make this idea possible, the measure distance must be absolute and general. So you have to give a measure of distance between a knot and the whole cloud.

The conventional parameterization methods described above are inadequate to generate a knot vector and respect the position and the relation including the internal correlation of all the rest knots. *The parameter value at each data point is chosen to be the parameter value related to all the rest.* The statistical distance between two points $x = (x_1, ..., x_p)^p$ and $y = (y_1, ..., y_p)^p$ in the p-dimensional space $\mathbb{R}^p$ is defined as [2,11]

$$d_s(x, y) = \sqrt{(x - y)^t S^{-1}(x - y)}\ . \tag{13}$$

and

$$d_s(x, 0) = \|x\|_s = \sqrt{x^t S^{-1} x}\ . \tag{14}$$

is the norm of x where S is the covariance matrix. This distance is also called quadratic distance. It measures the separation of two groups of objects. The data of the two groups must have the same number of variables (the same number of columns) but not necessarily to have the same number of data (each group may have different number of rows). *That's why we thought to use the distance of a knot to the rest of all cloud.* In fact this distance based on correlations between variables by which different patterns can be identified and analyzed. It differs from Euclidean distance by taking into account the correlations of the data set and the property of scale-invariant. In other words, it is a multivariate effect size.

Example [14]. Suppose we have two groups of data (Figure 1), each of group consists of two variables (x, y). The scattered plot of data is shown below:

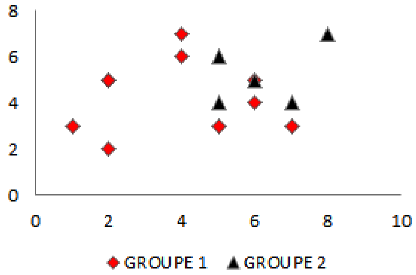


Fig. 1. Example of cloud of points

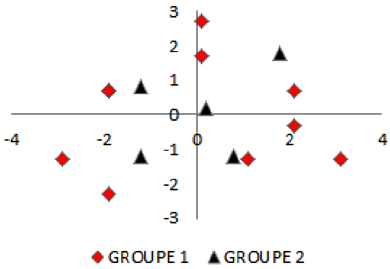


Fig. 2. Centered data

Group **G1** constituted with 10 points$(x_1, ..., x_{10})$ =[(2,2),(2,5),(6,5),(7,3),(4,7),(6,4),(5,3),(4,6),(2,5),(1,3)] respectively and group **G2** are also constituted with 5 points $(y_1, ..., y_5)$ = [(6,5),(7,4),(8,7),(5,6),(5,4)] respectively

1. First, we center the data figure (2) on the arithmetic mean of each variable. Covariance matrix of group i is computed using centered data matrix $\widehat{X}$

$$C_i = \frac{1}{n_i} \widehat{X}^T \widehat{X} \ . \tag{15}$$

   It produces covariance matrix for group 1 and 2 as follow

$$\begin{array}{cc} & \begin{array}{cc} x1 & y1 \end{array} \\ \begin{array}{c} x1 \\ y1 \end{array} & \begin{pmatrix} 3.89 & 0.13 \\ 1.36 & 0.56 \end{pmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{cc} x1 & y1 \end{array} \\ \begin{array}{c} x1 \\ y1 \end{array} & \begin{pmatrix} 0.13 & 2.21 \\ 0.56 & 1.36 \end{pmatrix} \end{array} \tag{16}$$

Fig. 3. Covariance Matrix for group 1 and 2

2. The pooled covariance matrix of the two groups is computed as weighted average of the covariance matrix. The weighted average takes this form :

$$s_i(r,c) = \frac{1}{n} \sum_{i=1}^{g} n_i c_i(r,s) \ . \tag{17}$$

   The pooled covariance is computed using weighted average (10/15)*Covariance group 1 + (5/15)*Covariance group 2 yields :

$$\begin{array}{cc} & \begin{array}{cc} X & Y \end{array} \\ \begin{array}{c} X \\ Y \end{array} & \begin{pmatrix} 3.05 & 0.27 \\ 0.27 & 1.93 \end{pmatrix} \end{array}$$

Fig. 4. Pooled covariance matrix

$$\begin{array}{cc} & \begin{array}{cc} X & Y \end{array} \\ \begin{array}{c} X \\ Y \end{array} & \begin{pmatrix} 0.332 & -0.047 \\ -0.047 & 0.526 \end{pmatrix} \end{array}$$

Fig. 5. Inverse Pooled covariance matrix

$$MeanDifference(G1 - G2) = (-2.3, -0.9) \tag{18}$$

3. The relation is simply quadratic multiplication of mean difference (18) and inverse of pooled covariance matrix (Figure 5).
4. When we get mean difference, we transpose it, and multiply it by inverse pooled covariance. After that, we multiply the result with the mean difference again and we take the square root. The final result is **Distance**(G1,G2)=1.41

As shown in the example, the correlation between all points comes into consideration in the calculation of this distance. In fact, this distance will be used to measure the distance between one point to another (including itself) instead of between two groups of points. So,we define the **new method** as:

$$d_i = Distance(D_i, E), \quad \text{i=1,...,m} \tag{19}$$

where **E** is the cloud of all the elements of U and D_i is the i^{th} control point. **Distance** is the distance set out above, d_i is the i^{th} distance between the control point D_i and all the cloud.

Algorithm 1. KnotVectCorrelation

```
   Input: n,p,E
   Output: U
1  begin
2      foreach control point D in E do
3          $xDiff = D - mean(E)$ ;
4          $cD = covariance(D)$;
5          $cE = covariance(E)$;
6          $pC = \frac{1}{(n+1)*cD} + \frac{n}{(n+1)*cE}$;
7          $d = \sqrt{xDiff * inverse(pC) * xDiff^t}$;
8      foreach i from 1 to n do
9          $t_i = t_{i-1} + \frac{d_{i-1}}{\sum_{k=0}^{n-1} d_k}$;
10     foreach j from 1 to n − p do
11         $U_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} t_i$;
```

KnotVectCorrelation is a simplified algorithm explaining our approach. Beginning with the *Input* :

- n : number of control point
- p : is the degree of curve
- E : is the cloud of all control points where D_i is the i^{th} control point

and the *Output* :

- U : Knot vector $U = \{u_0, u_1, ..., u_{n+p+1}\}$ with the condition $u_i < u_{i+1},\ i = 0, ..., n+p+1$ is well satisfied.

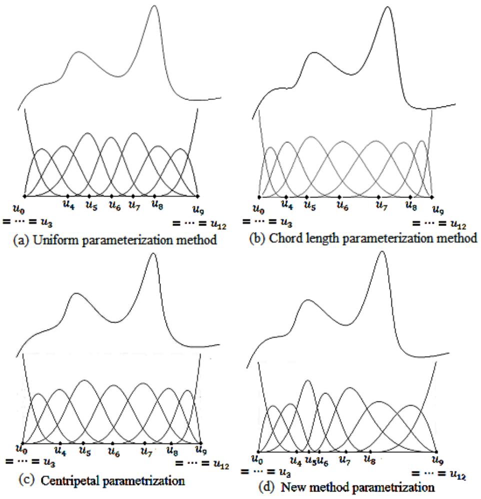
For line No.3 of the algorithm, mean(E) is the function that computes the mean difference explained above, t_i is the parameter value.

The averaging method for generating a knot vector is the most recommended, since a component of a knot vector is calculated (7) and the parameter value t_i (12) where the d_i (19), is to " average " the parameter which, is naturally well-organized by the new method.

This method is suitable when the points are divided horizontally. Especially, when the oscillations are strongly present, it remains much as it deals with issues, that are a little further, as noise.

4 Experimental Results

The experimental results of the new parameterization method are shown in the following figures. Uniform parameterization method attempts to fairly partition the nodes of the knot vector(Figure 6). Therefore, this method generates a lot of problems especially when the checkpoints are numerous. The major disadvantage is that the calculation of a node depends only on the position of the node above. We can see imperfections when the cloud is large and contains very frequent oscillations. The new approach tries to give the best approximation. In fact, the calculation of the current node involves the position of all nodes and the correlation of the latter with all the cloud.



(a) Uniform parameterization method

(b) Chord length parameterization method

(c) Centripetal parametrization

(d) New method parametrization

Fig. 6. The effect of applying the different methods of parameterization on the basis functions; Degree = 3

Figure 6 shows that the basis functions of this new method $N_{3,3}, N_{4,3}$ and $N_{5,3}$ are slightly higher than the others since it corresponds to the interval $[u_0$, $u_7]$ $[u_6$, $u_8]$ and $[u_5$, $u_9]$. These intervals are generated to minimize the difference between the cloud of points and the points that seem foreign. Note that the new method is used to slightly increase the oscillation (Figure7) of the 4^{th} and 6^{th} control points (Denote that the oscillation associated to a control point is the part of the curve which represents the desired approximation at this point i.e. the part $[u_i, u_{i+m}]$ for which the basis functions associated with this control point are nonzero.) thanks to the distribution of the knot vector. On the contrary, it does not do same treatment on the 5^{th} control point, since all the points with a downward swing is not far from the cloud of points.

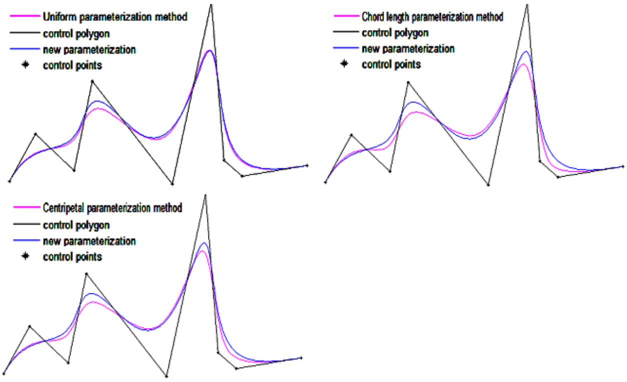


Fig. 7. Superposition of the two curves approximation

The superposition of the new method with existing methods shows that it does not always try to reduce the oscillations because this treatment does not make sense. Indeed this method treats the current swing, "imagining" the overall curve's shape and this of course through the distance (19). The comparison of the example in Figure 6 to that of Figure 7 shows that the new method reduces the oscillation corresponding to the 7^{th} control point since the majority of points are distributed horizontally. On the other side, in the example of Figure 6, it increases the oscillations of the 4^{th} and 6^{th} points since the cloud oscillate almost in the same way either the top or bottom.

Figure 8 shows that in the 7^{th} and 8^{th} control points the new method does not increase the oscillation as the majority of the control points are distributed linearly and considers these two issues as noise or foreign points. Thus, this method serves to minimize the error between the original curve and the obtained curve.

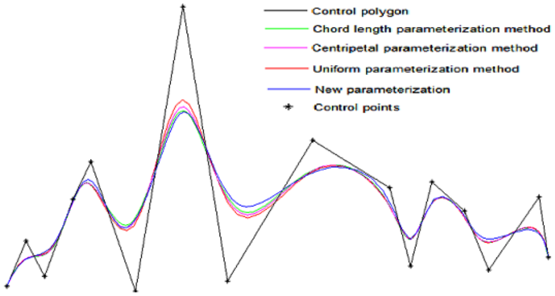


Fig. 8. Influence of the distribution of points to the curve using the new method

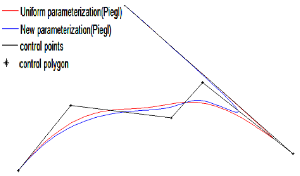


Fig. 9. Example of the new method using Piegl generation

This method gives a fine approximation curves using Piegl (Figure 9) in the generation of the knot vector.

5 Conclusion

NURBS curves and surfaces allow modeling a dozen of well-known basic objects (e.g. circle, ellipse and conic segments, sphere, etc.) and also used to describe free form objects. We described in this paper, a novel parameterization method of a knot vector in NURBS approximation. This new algorithm, based on interest of the correlation between control points. We use the notion of distance in NURBS curves approximation to solve the problems of oscillation. This novel method has produced good results for special type of curves and generally are widely used for approximation. It rivals the fabulous recent methods such as the hybrid and the centripetal. But there are still some minor problems. In fact, when the internal curve's curvature remains almost unchangeable, the new method addresses the problem, the same way, as other methods : we can therefore use another NURBS parameter, such as the weight.

References

1. Jung, H.B., Kim, K.: A New Parameterization Method for NURBS Surface Interpolation. Int. J. Adv. Manuf. Technol. 16, 784–790 (2000)
2. Joken, S., Naoya, I.: Geometrically local isotropic independence and numerical analysis of the Mahalanobis metric in vector space. Pattern Recognition Letters, 709–716 (2010)
3. Mohamed, N., Atef, H., Sawssen, J., Christiane, W.: NURBS Skeleton: A New Shape Representation Scheme Using Skeletonization and NURBS Curves Modeling, pp. 197–205. Springer, Heidelberg (2011)
4. Shamsuddin, S.M., Ahmed, M.A.: A Hybrid Parameterization Method for NURBS. In: Proc. of the international Conference on Computer Graphics, Imaging and Visualization (CGIV 2004), pp. 15–20. IEEE Computer Society, Washington, DC (2004)
5. Farin, G.: From Conics to NURBS: A Tutorial and Survey. IEEE Comput. Graph. Appl. 12, 78–86 (1992)
6. Choong-Gyoo, L.: A universal parameterization in B-spline curve and surface interpolation. Computer Aided Geometric Design 16, 407–422 (1999)
7. Lee, E.T.Y.: A Treatment of Conics in Parametric Rational Bezier Form, Boeing document, Boeing, Seattle, Wash. (February 1981)
8. Adi, D.I.S., Shamsuddin, S.M., Hashim, S.Z.M.: NURBS Curve Approximation using Particle Swarm Optimization. In: Seventh International Conference on Computer Graphics, Imaging and Visualization, pp. 73–79. IEEE Computer Society (2010)
9. Piegl, L., Tiller, W.: The NURBS book, 2nd edn. Springer-Verlag New York, Inc. (1997)
10. Godse, A.P., Gods, D.A.: Computer Graphics And Multimedia (2009)
11. Genichi, T., Rajesh, T.: The Mahalanobis-Taguchi strategy: a pattern technology system (2002)
12. Daviv, F.R.: An Introduction to NURBS. An Imprint of Elsevier (2001)
13. Lee, E.T.: Choosing nodes in parametric curve interpolation. Comput., 363–370 (1989)
14. Teknomo, K.: Similarity Measurement (2006)
15. Hoschek, J., Lasser, D.: Fundamentals of Computer Aided Geometric Design. A.K. Peters, Ltd. (1993)