Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (12): 2139-2145, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2139.2145

A Fast NURBS Interpolation Method for 3D Ship Hull Surface

[1]Guoyou Shi, [2,3]Shuang Liu and [4]Peng Chen
[1]Navigation College, Dalian Maritime University, Liaoning Dalian 116026, China
[2]School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[3]School of Computer Science and Engineering, Dalian Nationalities University, Dalian 116605, China
[4]Department of Computer Science and Technology, Dalian Neusoft
University of Information, Dalian 116023, China

Abstract: To describe 3D ship hull surface precisely and provide simulation help for charging-discharging of awkward length cargo, this paper proposed NURBS (Non-Uniform Rational B-Spline) based method to reconstruct 3D ship hull surface. Based on the study of basic theory of NURBS curve and surface modeling, many algorithms for NURBS such as the highly effective fast algorithm for calculating B-spline basis function, NURBS curve degree elevation algorithm based on end point interpolation are realized. A new method is further proposed to reconstruct NURBS surface by transfiguring fore and aft cross section line according to the hull lines plan or two-dimensional offset table. This method makes transfiguration to the cross section line situated at the bulb-bow and bulb-stern of the vessel, so that the projection of transfigured cross section line on the middle vertical section will be a curve and of isometry with the corresponding fore and aft contour line in broad sense, then a rectangular lattice will be formed by unified processing of these transfigured cross section lines together with other cross section lines and eventually a single hull NURBS surface will be reconstructed by using the end interpolation method. A simulation system is developed based on the proposed method and applied to awkward and length cargo charging and discharging. Compared to traditional modeling methods, this new simulation system can provide quick and precise loading scheme for shipping business.

Key words: NURBS, ship hull surface, interpolation, b-spline, reconstruction

INTRODUCTION

To simulate the process of charging and discharging awkward length cargo with computer is very important to both cargo owner and shipping companies. First of all, the owner can determine whether their cargoes are safely charged onto the ship with computer simulation. Secondly, shipping companies take advantage of charging and discharging the awkward length cargo and train shipboard operations staff using the simulation system to ensure vessel safety. To design and implement such a simulation system, one of the main tasks is to construct high precision three-dimensional model of the ship hull. The common methods of ship hull surface modeling are Coons surface, Bezier surface, B-Spline surface and NURBS surface modeling method (Wang and Zou, 2007; Wu *et al*., 2002; Chen *et al*., 2005; Lu *et al*., 2008). There are many methods of three-dimensional surface reconstruction (surface fitting), such as multiquadric method (Liu and Gao, 2011), thin-plate spline method (Yoo, 2011), finite element method (Sharf *et al*., 2007; Cai and Li, 2003), PDE (Partial Differential Equations) method (Baran *et al*., 2012; Linz *et al*., 2006), the minimum energy method (Labatut *et al*., 2009; Yang *et al*.,2009), non-uniform rational B-spline method (Xie *et al*., 2012; Sun *et al*., 2003; Ci and Li, 2004) and so on. In the choice of fitting function, such capacities as polynomial expression, local control, ensuring accuracy, ensuring geometric characteristics (information) undistorted and expressing a variety of curves and surfaces should be taken in consideration. In all these fit functions, only the NURBS meets all the mentioned requirements. It belongs to the Bezier functions cluster and is compatible with other CAD software, so it can further extend the functionality of the software. Because of its C^2 continuity, the second-order partial derivatives (surface curvature) at any point can be easily drawn, making the original geometric image avoid losing its internal characteristics. Again, its mathematical expressions in affine coordinates system (refer to zoom, rotate, pan) remain unchanged. NURBS surface modeling method has become the main trend in the field of CAGD

Corresponding Author: Shuang Liu, School of Computer Science and Engineering, Dalian Nationalities University, No. 18 Liaohe West Road, Dalian Development Zone, Dalian 116605, China Tel: 86-411-87656180

because it can express both the free curves and surfaces and analytic curves and surfaces. Since the shape of ship hull surface is complex, we usually use the three kinds of curves of waterline, cross-sectional line (station line) and the vertical section line to represent the ship hull surface, while the other ship hull shape information can be obtained through the interpolation of the three mentioned kinds of curves. In this paper, a simulation system is developed using VC ++ language, OPENGL three-dimensional graphics library and the ACCESS database to reconstruct the three-dimensional ship hull surface based on NURBS method. In the software implementation process, we also considered such issues as the hull surface pieces partition rules and implementation methods and mesh generation methods for control points at ship bow and stern.

DEFINITION AND FUNDAMENTAL PROPERTIES OF NURBS

The prominent advantages of NURBS method are to represent ruled and free surfaces in a unified mathematical form while having weight factors which can affect the shape of curves and surfaces, making it more appropriate to control and implement the shape. And NURBS method is the direct extension of non-rational B-spline method in four-dimensional space. The majority of properties and their corresponding algorithms of non-rational B-spline curves and surfaces are also available for NURBS curves and surfaces, easy to inherit and develop.

B-spline basis functions: There are many kinds of equivalent definitions for B-spline, such as Clark definition, difference quotient definition of trimmed power function, recursive definition of DeBoor-Cox (De Boor, 1972) and so on. It's true that the expression forms of the several definitions are different, but they are intrinsically related and their nature is the same. In order to facilitate computer implementation, recursive definition of DeBoor-Cox is given here. Supposing $U=\{u_0,u_1,...,u_m\}$ is a non-decreasing sequence of real numbers, for example, $u_i \le u_{i+1}, i=0,1,2,\cdots,m-1$. Here, u_i is called nodes and U is called a node vector. The i-th of p degree (p +1 degree) B-spline basis functions $N_{i,p}(u)$ is defined as:

$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u)$$
$$N_{i,0}(u)=\begin{cases}1 & u_i \le u < u_{i+1}\\ 0 & \text{otherwise}\end{cases} \quad (1)$$

Note that if the denominator of a fraction is 0, then the fraction value is taken as 0 in the above equation. Degree p of a given polynomial and node vector U can be used to calculate B-spline basis functions $Ni_{,p}(u)$. Although, the B-spline basis functions are defined on the whole parameter u axis, $Ni_{,p}(u)$ $\mu\in(u_i, u_{i+p+1})$ has a value greater than 0 only in the support interval, while the values out of the support interval are all 0. In terms of any one node interval, there are up to p+1 non-zero $Ni_{,p}(u)$, which means that $N_{i-p,p}, N_{i-p+1,p}, \cdots, N_{i,p}$ are non-zero. The simulation system developed in this study completed the fast algorithms of B-Spline basis function and its derivatives, which lays foundation for the following direct computations and inverse computation of NURBS curves and surfaces. The cubic B-spline basis function with 13 control points is shown in Fig. 1.

Non-uniform rational B-spline curves and surfaces: The p-degree NURBS curve is defined as:

$$P(u)=\sum_{i=0}^{n}R_{i,p}(u)d_i$$
$$R_{i,p}(u)=\frac{N_{i,p}(u)w_i}{\sum_{i=0}^{n}N_{i,p}(u)w_i}, \quad a \le u \le b \quad (2)$$

where, d_i are control points which constitutes control polygon. And w_i are the corresponding weights. $N_{i,p}(u)$ is normalized B-spline basis functions of degree p and defined on the non-periodic and non-uniform knot vector U. Unless it is otherwise indicated, a is equal to 0 and b is equal to1:

$$U=\{\underbrace{a,a,\cdots,a}_{p+1},u_{p+1},u_{p+2},\cdots,u_{m-p-1},\underbrace{b,b,\cdots,b}_{p+1}\} \quad (3)$$

The reason why the first and the end nodes in the node vectors are repeated for p+1 times is to guarantee the first and last data point of the curve consistent with

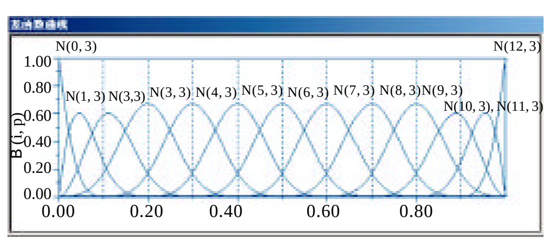

Fig. 1: B-spline basis function

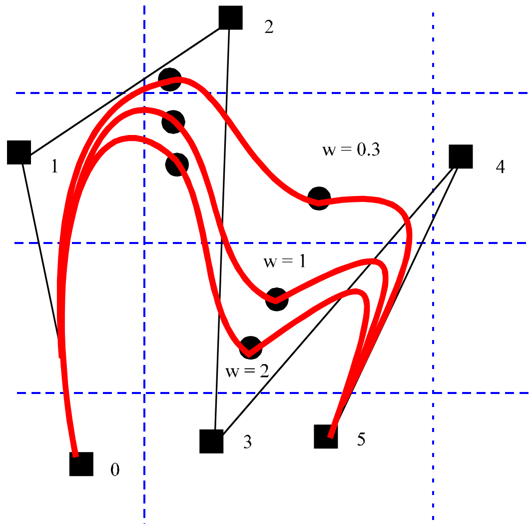

Fig. 2: Different weighted NURBS curves

the first and last control points. $R_{i,p}(u)$ resembles $N_{i,p}(u)$ in property. Apparently when $w_i = 1$ (i = 0,1, ..., n), a NURBS curve of k degree will degenerate into a B-spline curve, which means that the B-spline curve is a special case of NURBS curves. NURBS curves has properties such as locality, variation diminishing resistance, convex hull property, the invariance under affine and perspective transformation, parametric continuity as well as shape-regulation of weight factors, etc. Weight factors have clear geometric meaning. Fixing all the control vertices, when a weight factor change but the other weight factors remain constant, the greater the value of the weight factor is, the closer the curve apart from control points is; conversely, the smaller the value the weighting factor is, the further the curve away from the control vertices is, as shown in Fig. 2.

A NURBS surface of p degree in u-direction and q degree in v-direction is defined as:

$$P(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)d_{i,j}$$
$$R_{i,j}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \quad (4)$$

where, d_{ij} are the control points and W_{ij} are the weights. $Ni_{i,p}(u)$, $Ni_{j,q}(v)$ are the B-spline basis functions defined on non-periodic and non-uniformed knot vectors U and V:

$$U = \{\underbrace{0,0,\cdots,0}_{p+1}, u_{p+1}, u_{p+2}, \cdots, u_{r-p-1}, \underbrace{0,0,\cdots,0}_{p+1}\}$$
$$V = \{\underbrace{0,0,\cdots,0}_{q+1}, u_{q+1}, u_{q+2}, \cdots, u_{s-q-1}, \underbrace{0,0,\cdots,0}_{q+1}\} \quad (5)$$
$$r = n + p + 1, \quad s = m + q + 1$$

In practice, what we usually use is cubic NURBS curves and surfaces.

CUBIC NURBS CURVE AND SURFACE INTERPOLATION CACULATION

With the known control vertices d_i and node vector U, the process of finding points P(u) on the curve is called direct computation in NURBS. On the contrary, knowing a series of points P_i and knot vector U on the curve, we call the process of seeking control vertices d_i the inverse computation in NURBS. Mathematically speaking, direct computation is called approximation, which is to construct a curve to approximate the polygon consist of the known sequence of points and, generally curves do not go through the given points. While inverse computation in mathematics is called interpolation: that is to construct a curve to interpolate at given points which is so-called point-point passing. Both positive and inverse computation operators require determining the knot vector first to compute B-spline basis functions. Knot vectors are usually decided by the sequence of points on the known curve, the interpolation points, or the curve control points (Fa Farin, 1997).

Determination of cubic NURBS curve knot vector: The identification of knot vectors can be considered as two cases: One is to find the knot vectors in the case of knowing the series of points on the curve called interpolation points P_i (i = 0,1, ..., n) and another is to find the knot vector in the case of knowing the polygon control points on the curve. This paper describes the method of constructing knot vectors based on interpolation points. In order to make a NURBS curve of three degree passing through a set of data points P_i (i = 0,1, ..., n), its inverse computation process generally makes the first and last points of the curve consistent with the first and last data points respectively. Data points P_i (i = 0,1, ..., n) will correspond to the nodes the domain of cubic NURBS curves in turn, meaning that the P_i has node value u_{i+p}, where the degree p of NURBS curves is 3. The cubic NURBS interpolation will be defined by n+3 control vertexes d_i (i = 0,1, ..., n +2), so the corresponding node vectors are $U = (u_0, u_1, ..., u_{n+6})$ and the domain of the curve is $u \in (u_3, u_{n+3})$ and $u_0 = u_1 = u_2 = u_3 = 0$ and $u_{n+3} = u_{n+4} = u_{n+5} = u_{n+6} = 1$,. To determine the parameter value u_{i+3} (i = 0,1, ..., n) corresponding to the data point P_i (i = 0,1, ..., n), it's necessary to process parameterization of the data points. There are several methods of parameterization as follow: (1) Uniform parameterization; (2) Centripetal parameterization; (3) Clipped chord length parameterization. The method of accumulation of chord length parameterization, used in this study, makes each node interval length correspond to the chord length of the correspondence curve then the node parameters are determined as follows:

$$u_0 = 0, \quad u_i = u_{i-1} + |\Delta P_{i-1}|, \\ \Delta P_{i-1} = P_i - P_{i-1}, \quad i = 1, 2, \cdots, n \tag{6}$$

Here, ΔP_{i-1} refers to the forward difference component. Let L be the total length of the chord, then:

$$L = \sum_{i=1}^{n} |P_i - P_{i-1}|$$

We can get the following equation after being normalized:

$$u_0 = 0, \quad u_i = u_{i-1} + \frac{|P_i - P_{i-1}|}{L}, \ i = 1, 2, \cdots, n \tag{7}$$

This method honestly reflecting the distribution of data points according to the chord length has always been considered as the best parameterization method. It resolves the problems that have arisen in the case of using the uniform parameterization under the uneven distribution of data points. In most cases it can give us a satisfactory results and the resulting interpolation curve has relatively good smoothness.

Interpolation calculation of cubic B-spline: Given a group of data points Pi (i = 0,1, ..., n) and the corresponding node values □□u_i (i = 0,1, ..., n), node values are generally calculated based on the above section rather than given. Now what we want to find is a B-Spline curve s of three degree determined by the nodes u_i and unknown control points di (i = -1,1, ..., n +1), making $s(u_i)$ equals to P_i and transforming a B-Spline curve of three degree into segments Bessel (Bezier) curves b_i shown as:

$$\begin{aligned} & P_i = b_{3i}\ ; i = 0, \cdots, n \\ & P_i = \frac{\Delta_i b_{3i-1} + \Delta_{i-1} b_{3i+1}}{\Delta_{i-1} + \Delta_i}, i = 1, \cdots, n-1; \\ & b_{3i-1} = \frac{\Delta_i d_{i-1} + (\Delta_{i-2} + \Delta_{i-1}) d_i}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i}, i = 2, \cdots, n-1; \\ & b_{3i+1} = \frac{(\Delta_i + \Delta_{i+1}) d_i + \Delta_{i-1} d_{i+1}}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}}, i = 1, \cdots, n-2; \\ & \Delta_i = \Delta u_i = u_{i+1} - u_i\ ; \Delta_{-1} = \Delta_n = 0 \end{aligned} \tag{8}$$

Cases for end points are:

$$\begin{aligned} & b_2 = \frac{\Delta_1 d_0 + \Delta_0 d_1}{\Delta_0 + \Delta_1} \\ & b_{3n-2} = \frac{\Delta_{n-1} d_{n-1} + \Delta_{n-2} d_n}{\Delta_{n-2} + \Delta_{n-1}} \end{aligned} \tag{9}$$

We can get the following equations based on the above:

$$\begin{aligned} & (\Delta_{i-1} + \Delta_i) P_i = \alpha_i d_{i-1} + \beta_i d_i + \gamma_i d_{i+1} \\ & \alpha_i = \frac{(\Delta_i)^2}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i} \\ & \beta_i = \frac{\Delta_i (\Delta_{i-2} + \Delta_{i-1})}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i} + \frac{\Delta_{i-1} (\Delta_i + \Delta_{i+1})}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}} \\ & \gamma_i = \frac{(\Delta_{i-1})^2}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}} \end{aligned} \tag{10}$$

Select two Bezier points b_1 and b_{3n-1} according to the boundary conditions and set d-1equals to P_0 and d_n+1 equals to P_n and we can get the following Eq:

$$\begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & & & \\ & & & \ddots & & \\ & & & \alpha_{n-1} & \beta_{n-1} & \gamma_{n-1} \end{bmatrix} \begin{bmatrix} d_1 \\ \vdots \\ d_{n-1} \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_{n-1} \end{bmatrix}, r_i = (\Delta_{i-1} + \Delta_i) P_i\ , i = 1, \cdots, n-1 \tag{11}$$

The Bezier and B-Spline boundary conditions can be determined by the following boundary tangent vector:

$$\begin{aligned} & b_1 = r_0 = \frac{2(2\Delta_0 + \Delta_1)}{\Delta_0 \beta_0} P_0 + \frac{\beta_1}{2\Delta_0 \Delta_1} P_1 - \frac{2\Delta_0}{\Delta_1 \beta_1} P_2 \\ & b_{3n-1} = r_n = \frac{2\Delta_{n-1} + \Delta_1)}{\Delta_{n-2} \beta_{n-1}} P_{n-2} - \frac{\beta_{n-1}}{2\Delta_{n-2} \Delta_{n-1}} P_{n-1} + \frac{2(2\Delta_{n-1} + \Delta_{n-2})}{\beta_{n-1} \Delta_{n-1}} P_n \end{aligned} \tag{12}$$

The key of surface inverse computation-interpolation is to solving Eq. 11 and then the key to solving equations is to determine the appropriate boundary splines. Next, we discuss the various bound ary conditions, respectively. (1) Tangent vector condition: the first and the last tangent vectors of the curve respectively influence the control vertices d-1 with d_0 and d_n and $d_{n}+1$; the first and last control vertices are the first and the last data points, so the first and the last tangent vectors respectively influence the control vertices d_1 and d_{n+1}. There is only data points without given endpoint conditions when constructing the hull surface. In this case we can construct it through the numerical differentiation based on the given data points. (2) Free endpoints condition: Let the endpoint second derivative vector equal to a zero vector. (3) Parabola condition: In practice, the boundary condition cannot be used as an independent method, but it can construct a parabola based on the given data points, then determine the ends tangent vectors supplied to the tangent vector conditions.

Inverse computation of control points for cubic NURBS interpolation curve: The problem of non-uniform rational B-spline interpolation is as follow. Given a set of data points P_i (i = 0, 1, ..., n) and the corresponding node values □□u_i (i = 0, 1, ..., n) and the corresponding weight factors w_i (i = 0, 1, ..., n), then what we should do is to find a NURBS curve s of degree 3 determined by the nodes u_i and unknown control points d_i (i = -1, 1, ..., n +1) and the corresponding weights v_i to satisfy Eq. 2. The solution is to input the output 3-dimensional data into the Eq. 11. But the data used here is four-dimensional, so we can make interpolation of data points (wi Pi,)in four-dimensional space, then map the control polygon point (w_i P_i, w_i) obtained in four-dimensional space to the three-dimensional space and finally get the control points (e_j, v_j) of NURBS curve in three-dimensional space, where v_j is the weight of control point d_j. Here weights w_i are not given, so it is necessary solve it before solving linear equations. There is no best way on how to select the weight factor of given data points. Sun (1998) got an optimal solution with Wolfe algorithm by constructing quadratic programming problems. Wang and Li (2001) automatically calculated a set of initial weights of the NURBS curve using linear interpolation and embedded ANN (Artificial neuron network) algorithm, based on the shape of the NURBS control polygon and curve shape constraints. If all weights are taken as 1, then the NURBS curve is consistent with full (non-rational) B-spline. Since B-spline basis function is non-negative, so w_i can be assumed greater than or equal to 0. If w_i-1 and w_i+1 are fixed, NURBS curve will leaves from the control points when w_i makes decreasing variation in the (0, 1) interval. We will get the opposite result when w_i makes incremental change in (1, 8) interval. When w_i equals to 0, the corresponding control points will be ignored. This paper calculates the initial weights w_i (i = 0, 1, ..., n) taking the interpolation points as control polygon vertices using the second method. This method requires that the constraints of each control vertexes are prior given, especially at the type lines of ship bow and stern, where we must be very careful when setting the constraints in sharp places with weight values big enough. If the curve obtained from the interpolation is not ideal, we can regulate the constraints according to the shape of the curve until we get a satisfying NURBS curve.

INVERSE COMPUTATION OF CUBIC NBURS SURFACE AND COMPUTER SIMULATION

Key of NBURS constructing ship hull surface is to obtain the given topology rectangular array of data points $P_{i,j}$ (i = 0,1, ..., m; j = 0,1, ..., n) and the corresponding control points of $d_{i,j}$ (i = 0,1, ..., m + p-1; j = 0,1, ..., n + q-1) interpolated by a NBURS surface of p*q degree. The common approach is to transform the problem of surface inverse computation into curves inverse computation of two degree. With the data points meshes of hull surface known, NURBS surfaces are constructed through the given data points, called surface interpolation, which is to find the control points mesh of the surface knowing the NURBS surfaces, also known as the inverse computation problem. NURBS curve interpolation method can be directly extended to NURBS surface interpolation which includes two steps: (1) Interpolating into the given data points P_{ij} and their corresponding weights along u(or v) to find the corresponding control points and its weight factors; (2) Taking the obtained control vertexes and their weights as the new data points and the weighting factors and completing interpolation to obtain the control vertexes and weights of NURBS surfaces along v (or u).
Ship hull interpolation surface is usually constructed based on data points of offset table and the points given in the offset table are relatively regulated and showing substantially topology rectangular. Provided that along the direction of waterline is *u* direction and the direction of cross-sectional line is *v* direction, then the cross-sectional line and the waterline will interwoven into meshes to form the ship hull surface. But the number of points on each cross-sectional line and water line is not exactly the same. As many of these points cannot constitute a regular rectangular topology grid, they need to be processed. In this paper, we use the method of dividing each line in average to obtain a regular grid. According to the offset table, we make inverse computation calculation of each waterline and cross-sectional line to obtain the interpolation waterline and the interpolation cross-sectional line at first. Divide every interpolated cross-sectional line and interpolated waterline into n parts in average to obtain uniform knot vectors of vu direction, making the distribution of data

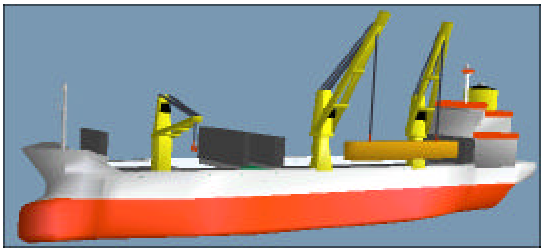

Fig. 3: 3D ship surface reconstructed by NURBS

point's grid more evenly. Then take the equally divided node vectors as the new the new node vector and re-do interpolation calculation to get the new interpolation cross-sectional lines and new interpolation waterlines of the new uniform knot vectors and finally get the whole bi-cubic ship hull B-spline surfaces. Based on this, we finished a highly realistic dynamic simulation system of loading and unloading the awkward length of cargo using VC++ language and OPENGL graphics library. The result of computer simulation is shown in Fig. 3.

In this example, we chose the boundary conditions for the boundary tangent vectors making the slope at the intersection of the curve and the straight line the same while achieving smooth connection between curve and straight line. For a closed curve, we used the method of making the periodic B-spline closed curve and open curve unified to represent it. After the data points are given, the control points are not only related to data points, but also to the boundary conditions, the value point of parameterization and weighting factor. The boundary conditions can only influence the curve shape at both ends of the curve and parameterization can influence the whole shape of the curve, but a change in weight factor influences only the shape of several paragraphs of the curve related to the corresponding control points.

CONCLUSION

With NURBS surface model we can finish two tasks. The first one is to make surface approximation. Given initial control vertices at first determining the knot vectors of v, u parameter directions and then constructing a surface approximating the control vertices is called direct computation. The other one is surface expression. Given some of the data points on the surface at first and the final surface constructed must be through these data points, namely inverse computation. For the given line offsets of the hull surface, we use the non-uniform rational B-spline interpolation of ship hull surface to control the boundary conditions of the surface. Through interpolation and other corresponding transformation processing, we can get the whole ship hull surface parametric equations. With the help of computer we can solve a large number of practical problems such as ship overall performance computing (hydrostatic force calculation, stability calculations, anti-sinking calculation, ship cabin capacity calculation, ship hull wetted area calculation, etc.). In addition, this provides a new approach to creating an accurate and highly realistic three-dimensional hull model in the large ship maneuvering simulator.

REFERENCES

Baran, M.S., R.L. Tutwiler and D.J. Natale, 2012. Surface reconstruction for 3D remote sensing. Proc. SPIE Inter. Soc. Optical Eng., 8399: 15-15.

Cai, Z.Y. and M.Z. Li, 2003. A smoothing-finite element method for surface reconstruction from arbitrary scattered data. J. Software, 14: 838-844.

Chen, J., H.G. Cui and J.J. Liu, 2005. Study and implementation of complex ship-hull surface?s NURBS modeling. Comput. Eng., 31: 201-202.

Ci, R.M. and D.B. Li, 2004. NURBS surface reconstruction in reverse engineering. J. Nanjing Unive. Sci. Technol., 28: 390-394.

De Boor, C., 1972. On calculating with b-splings. J. Approximation Theory, 6: 50-62.

Farin, G., 1997. Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide. 4th Edn., Academic Press, London.

Labatut, P., J.P. Pons and R. Keriven, 2009. Robust and efficient surface reconstruction from range data. Comput. Graphics Forum, 28: 2275-2290.

Linz, C., B. Goldlucke and M. Magnor, 2006. A point-based approach to PDE-based surface reconstruction. Pattern Rec., 4174: 729-738.

Liu, J., D. Liu and F. Gao, 2011. The application of multiquadric function in surface reconstruction. Proceedings of the International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), August 19-22, 2011, Jilin, pp: 895-897.

Lu, C.H., Y. Lin, Z.S. Ji and M. Chen, 2008. NURBS based ship form design using adaptive genetic algorithm. Proceedings of the International Offshore and Polar Engineering Conference, July 6-11, 2008, Vancouver, BC, Canada, pp: 535-540.

Sharf, A., T. Lewiner, G. Shklarski, S. Toledo and D. Cohen-Or, 2007. Interactive topology-aware surface reconstruction. ACM Trans. Graphics, Vol, 26, 10.1145/1275808.1276431

Sun, J.G., 1998. Computer Graphics. Tsinghua University Press, Beijing.

Sun, Y.W., Y.C. Wang and W.J. Liu, 2003. Free-form surface reconstruction based on nurbs to serial cross-sections. Chinese J. Mech. Eng., 16: 420-423.

Wang, H. and Z.J. Zou, 2007. Geometry modeling of ship hull based on NURBS. Ship Technol. Res., 54: 145-148.

Wang, X.B. and S.Y. Li, 2001. Automatic calculation of initial weights for NURBS. Acta Aeronautica et Astronautica Sinica, 22: 184-186.

Wu, D.W., Y. Lin and Z.S. Ji, 2002. Ship hull surface?s NURBS representation and design. J. Dalian Unive. Technol., 42: 569-573.

Xie, W.C., X.F. Zou, J.D. Yang and J.B. Yang, 2012. Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. Comput. Aided Design, 44: 1127-1140.

Yang, J., C.Q. Zhu and Q. Xing, 2009. Algorithm for visualization of implicit surfaces reconstructed from point cloud data. J. Comput. Appli., 29: 1901-1905.

Yoo, D.J., 2011. Three-dimensional surface reconstruction of human bone using a BB-spline based interpolation approach. CAD Comput. Aided Design, 43: 934-947.