Computer-Aided Design and Applications

http://www.cadanda.com

Reconstruction of Lines in a Ship Hull with B-Splines

Francisco Pérez

Polytechnic University of Madrid, francisco.perez.arribas@upm.es

ABSTRACT

A method for reconstructing lines from a point cloud with the use of B-splines is described in this paper. The method operates directly on the point cloud without any explicit or implicit surface reconstruction procedure, and works with unordered points. The presented technique is based on a Least Squared approximation, an iterative method for the parameterization and the use of fairing conditions. The method is designed to work with scattered data points from a 3D scanner and applied in ship reconstruction, where some specific lines contained in the ship hull are required and the full reconstruction of the ship hull surface is unnecessary.

Keywords: least squares, reverse engineering, point clouds.
DOI: 10.3722/cadaps.2008.99-109

1. INTRODUCTION

With 3D scanners becoming the standard source for geometric data acquirement, the work with point clouds that these devices produce has received an increasing amount of attention: Point based techniques that use the self point cloud as an alternative surface representation [1], which includes point rendering [2], parametrization, thinning, and reconstruction.

Besides point based techniques, grid generation methods are normally used to fit a triangular mesh to the 3D point cloud, [3], and [4]. A decimation process is also required to reduce the number of triangles to those essentially needed to define the object and make it more manageable. A treatment of the generated mesh can produce NURBS patches to represent the scattered data, [5], and [6]. Some commercial software products exist in order to facilitate this non-trivial procedure, which may require spending a lot of time on data cleaning and data preparation and usually the use of large amounts of time and space if the number of points is gigantic.

In some cases, a complete surface reconstruction is not necessary but only in some particular areas of the surface reproduced from the point cloud, or in the case of a shipyard or a model basin, specific lines of a ship or of a model hull. Curve modeling continues to play an important role in ship hull design because designers and engineers still prefer the two step process of curve-net modeling, and surface creation.

If we want to make a shape/quality control of a constructed ship or model hull surface, we will try to obtain the hull stations (longitudinal sections perpendicular to the waterplane and to the centerplane) and compare these stations with the ones of the lines drawing, which includes these stations, waterlines (sections parallel to the water plane), and longitudinals (sections parallel to the ship's centerplane). The stations allow to make hydrostatic and stability calculations.

If we want to take out a ship's conventional bow in order to construct a new bulbous bow, the shipyard will make the split at a station (for structural and economical reasons) and therefore, the form of this specific station is needed. In a constructed ship, this is not easy to do because the ship hull is deformed due to normal operation and big deviances from the original lines drawing are expected. The installation of bilge keels on a hull is another example: these keels will be placed along the curved part of the ship hull and the base points on this hull are needed in order to facilitate the construction and installation. There are more examples where specific lines on a surface are needed and all these can be managed with the use of 3D scanners and the resulting work with the scattered data points.

In this paper, a method to extract some specific lines from a point cloud is presented. Instead of using a point based technique, a B-spline representation of these points is made. B-splines curves and surfaces are a common reference in the industry and permit the interchange of the reconstructed curves with specific software (Naval architecture programs, Finite analysis methods…) through standard files .dxf, .igs,…

The method works with unordered data points, and accepts a B-spline of an arbitrary degree. Ordering scattered curve points is not trivial and a direct approach of the problem with a common Least Squares (LS) approximation will result in failure as will be presented in the examples used in this paper. The B-spline fitting is made with special emphasis on the choice of parameterization, which is relevant to increase the accuracy of the splines and to impose a tolerance constraint. Finally, some extra conditions to the control points of the B-spline should be imposed, in order to obtain a realistic approximation of the data points. These conditions can be set in the definition of the LS system with the use of fairing equations.

A global outline of the method is made in section 2, which contains a description of the LS method, information about the parameterization, and how to include a fairness metric and conditioning in the definition of the presented algorithm. Furthermore, the way to include these three items in order to obtain an approximating B-Spline is described. A numerical example to make out the different effects of the mentioned aspects on scattered data is presented in section 3. The method is then tested in real scenarios of a ship model's bulbous bow, where some lines of the hull are reconstructed from its scanned data points. Finally, a number of conclusions are drawn.

2. DESCRIPTION OF THE METHOD

The starting point is a cloud of points (Fig. 1, left) obtained from a 3D scanner (Fig. 1, center). Because the objective is to extract certain lines contained in the scattered data, (i.e. in the case of Fig. 1 right, the stations and waterlines marked on the surface) the first step is to obtain the points corresponding to the lines. In the case of stations or waterlines, a binary search of the points placed between two parallel planes is enough and a "thick line" of points (Fig. 2) corresponding to this station or waterline is obtained. In the case of stations or waterlines, a nearly plane set of points is obtained but the method works well for a 3D set of points; for example points between two limiting cylinders that can be used to place a bow propeller tunnel in a constructed hull.

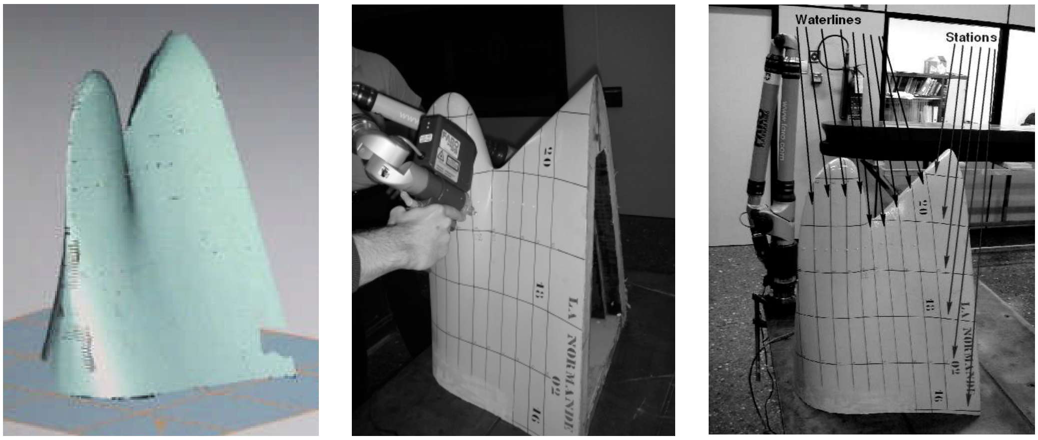


Fig. 1: Grid of points and physical object.

Once the points of the line of interest (data points) have been obtained, the method will apply a LS approximation to the data points (Fig. 2), which will not present an ordered sequence since they have not been acquired sequentially and this information will always contain some amount of noise. The parameterization of these scattered data points is essential in order to obtain good results and will be explained in detail further on. These parameters are an input for the LS approximation. In the choice of the parameterization, the shape and tolerance requisites will be imposed: these requirements will maintain the distance of the data points to the B-spline under a certain tolerance. Therefore, this distance is implicitly included in the way that the parameters of the points are obtained and as a result, the output of the method includes the average and maximum distance of the data to the B-Spline.

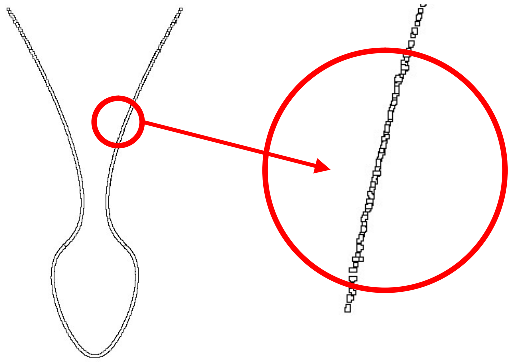


Fig. 2: Curve to be approximated.

A direct approach to the LS, even with the appropriate parameterization, will produce a mathematical solution of the LS equation system, which although mathematically correct is not a practical solution. The result that is obtained is unlikely to fit the shape requirements well, and surely will not be a faired or smooth curve (See section 3). Up to now, no consideration to the fairness of the B-spline has been included in the formulation of the problem.

In other words, the method first extracts the line of interest from the points of the whole surface, then obtains the parameters of the data points that takes into account a distance allowance, and finally applies a LS method that considers some fairing conditions. The method is an iterative procedure: if the tolerance is not attained, the user can increment the number of iterations to obtain the parameterization of the data, add more control points, or even increase the degree of the B-spline. Each step will now be analysed in detail.

2.1 The Least Squares Approximation

This subsection is also employed to set nomenclature. In this problem, the large number of data points and their scattering do not suggest the use of an interpolating B-spline. Thus, an approximating curve is needed. The B-spline s(u) will not pass through the data points exactly but will pass close enough to the points to capture the inherent shape of the points. This is the well known Least Squares approximation [7]. In this problem, n+1 data points $Q_0, \ldots Q_n$ will be approximated by a B-spline of p^{th} degree, with N+1 control points $P_0, \ldots P_N$, $N < n$, that are unknown and are obtained as the final result of the whole method. The general LS problem is described by the overdetermined set of n+1 equations (1) with N+1 unknown variables:

$$\begin{array}{l} B_0^p(t_0)\cdot P_0 + B_1^p(t_0)\cdot P_1 + \ldots + B_N^p(t_0)\cdot P_N = Q_0 \\ B_0^p(t_1)\cdot P_0 + B_1^p(t_1)\cdot P_1 + \ldots + B_N^p(t_1)\cdot P_N = Q_1 \\ \quad\vdots \qquad\qquad\quad \vdots \qquad\qquad\qquad \vdots \qquad\quad \vdots \\ B_0^p(t_n)\cdot P_0 + B_1^p(t_n)\cdot P_1 + \ldots + B_N^p(t_n)\cdot P_N = Q_n \end{array} \tag{1}$$

Where the B_i^p corresponds to the i^{th} basis function of a p^{th} degree B-spline, that is calculated using the de Boor's algorithm considering a uniform knot vector; t_j (j=(0,n)) represents the parameters of the data points, whose calculation will be explained in the next section. Matrix expressions are convenient to solve the problem:

$$\begin{bmatrix} B_0^p(t_0) & B_1^p(t_0) & \cdots & B_N^p(t_0) \\ B_0^p(t_1) & B_1^p(t_1) & \cdots & B_N^p(t_1) \\ \vdots & \vdots & \vdots & \vdots \\ B_0^p(t_n) & B_1^p(t_n) & & B_N^p(t_n) \end{bmatrix} \cdot \begin{vmatrix} P_0 \\ P_1 \\ \vdots \\ P_N \end{vmatrix} = \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_n \end{bmatrix} \Rightarrow [M]\cdot[P]=[Q] \Rightarrow [M]^T\cdot[M]\cdot[P]=[M]^T\cdot[Q] \tag{2}$$

This system of equations is solved by multiplying both sides of (2) by $[M]^T$ which creates a determined linear system of N+1 x N+1 dimension. These kind of systems can present very poor conditioning, especially if a large number of control points is used. If a conventional technique is used to solve this ill-conditioned system, problems can be expected. This can be improved if a single value decomposition of $[M]^T \cdot [M]$ and a later back-substitution process is made. The solutions of this system are the control points of the best B-Spline fitting.

2.2 The Parameterization

The Q_i points of Eq. (1) need a parameter so system (2) can be solved. An approach to this problem with a standard parameterization such as the centripetal, chord-length… does not consider the effect of the distance of the data points to the B-spline. Also, in case of disordered data, this will produce a non-realistic set of parameters. In this method, a parameterization based on a minimum distance is adopted. The process is iterative, and is described in these three points:

1. The method starts with a centripetal parametrization of the Q_i points and system (1) is solved. This produces an invalid B-Spline, but it will be used as a starting curve of the iterative process only for the first loop.

2. For each Q_i, the minimum distance to the B-Spline is calculated. This is done by dividing the B-Spline s(u) in Bézier curves $b_j(u_L)$ (j=1,N-p) of the same degree, and computing the minimum distance to the corresponding Bézier piece, that leads to solve equation (3).

   $$(Q_i - b_j(u_L))\cdot(b'_j(u_L)) = 0 \tag{3}$$

   This equation is solved in the local domain of the Bézier curves, and it is a polynomial equation. This means that $u_L \in [0,1]$ and specific algorithms for this type of equations can be used. These algorithms do not need an initial guess, which will be required if a Newton method is used in the B-Spline domain. In the presented method, a Jenkins-Traub 3-stage algorithm [8] is used and the valid solution will be a non-complex solution of $u_L \in [0,1]$. In case of multiple solutions that fulfill the mentioned conditions, the closest one to Q_i is the correct one.

   Once the solution has been found, the local u_L for the Bézier domain is easily converted into its global value t_i in the B-spline domain. This t_i value is the parameter associated to the point Q_i when solving the system (1).

3. Aside from obtaining the t_i (i=1,n) values, a distance $d_i = (Q_i - s(t_i))$ is computed. This is the Euclidean distance between Q_i and the B-Spline, and it is used to check the shape requirement. If the maximum distance d_i (i=1,n) is above a given tolerance, a new loop repeating steps 2 and 3 is made until the maximum distance is below this value. More specifically, the quality of the obtained curve is measured using this tolerance constraint: the shape of the B-Spline is amended with the use of this parameterization (Fig. 3) and the quality is evaluated based on this tolerance constraint.

   If this tolerance is not obtained in less than 50 iterations, it means that the number of control points N has to be increased. The increment of the degree p in this procedure reduces the maximum distance too, but increasing the number of control points has a larger effect. Besides, a higher degree raises the complexity and the computational time.

   Since the parameterization is iterative, for every Q_i an associated index k (k=1, N-p) referring to the Bézier piece where the minimum distance was achieved in the previous iteration loop, improves the speed of the calculations. In the step 2 of the next loop, Eq. (3) is just applied on k, or in its nearest piece (k-1 or k+1) just in case the change in the B-Spline shape alters the Bézier piece were the minimum distance is fulfilled, in respect to the previous loop.

The effect of this iterative parameterization in ordered data can be seen in Fig. 3. Ordered points of a ship's half-station of maximum dimensions of 6.35 m x 1.7 m are approximated by a cubic B-spline of N=6, with the centripetal parameterization shown in Fig. 3.1. The effect of a single iteration is shown in Fig. 3.2. The error or maximum distance between the points and the B-spline is reduced as the number of iterations increase, 20 in Fig. 3.3 and 50 loops in Fig. 3.4.

This works quite well in the case of ordered points, but a third stage has to be included to work with unordered data points. This third step is to include fairing conditions into the definition of the method.

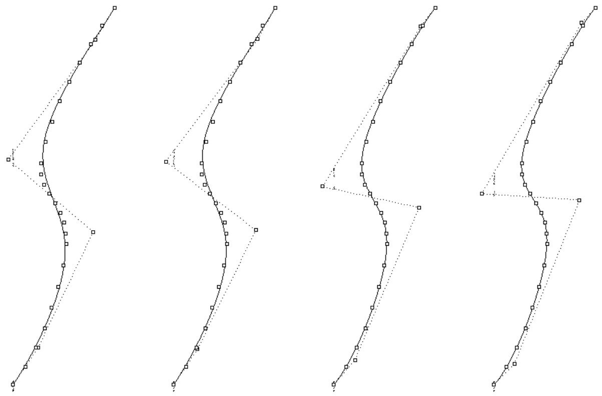


Fig. 3: Effect of the parameterization.

2.3 Fairness Conditions

Fairing is an important part of the design and manufacturing process. This kind of fitting techniques should produce a smooth or fairing approximating curve since the ship hull is a faired surface. The problem is the LS only minimizes the distance between the points and the B-spline, and does not consider the shape of the resulting curve. This means that the result of (2) will be a curve that may squirm or curl producing big disturbances in the curvature, although it is as close to the data points as a result of the LS technique.

Fairness in practical applications is a very subjective matter. The widely accepted definition states that a curve is fair if its curvature is continuous and consists only of a few monotone pieces. In ship hull design, where the ship lines are necessarily faired since the water flow can dispread out the ship's hull increasing resistance, a common definition is that a faired line has to minimize unnecessary variations of its curvature, since this will cause a viewer's eye to stop when inspecting the curve (and water flow to dispread). This is related with the first definition and an application of "the principle of simplest shape" of common use in design practice.

Many papers in CAD treat the problem of fairing (most of the times of planar fairing of cubic B-Splines) which can be classified in two categories. The first one uses local fairness criteria, such as the magnitude of the nodal discontinuities of the curvature (k) slope [9], [10]. The second category uses global fairness criteria and usually employs integral formulae through all the parameterization intervals. The integrand involves either the square of the curvature, the square of the second or a higher order derivative of the curve, or as a relative weight combination of the second and third derivatives [11].

The presented method uses a global approach to the problem, since a global definition of the curve is obtained from the LS. The fairing will be included trying to minimize a fairing function (4), based on weighted combinations of the second and third derivatives (a+b=1).

$$f = a \cdot \int_0^1 \left\| \frac{d^2}{du^2} s(u) \right\| du + b \cdot \int_0^1 \left\| \frac{d^3}{du^3} s(u) \right\| du \tag{4}$$

The problem arises because there is not a direct definition of the B-spline until the LS method is applied and it is not possible to include Eq. (4) directly into the definition of the LS fitting. The key is to change the focus from the curve itself to its control polygon, since it is inside the definition of the system (1) and is the final result of the LS fitting. Because of the properties of the B-Splines, the curve mimics the shape of its control polygon: if the polygon conducts well (does not wiggle or twist), so does the B-Spline.

Thus, instead of applying Eq. (4) directly, the method considers the minimization of the second and third derivatives included in (4) changing derivatives by differences of the control points. If the differences of the control polygon decrease, so do the derivatives of the B-spline. This can be accomplished setting the conditions of Eq.(5) to the control points for the second differences:

$$\begin{gathered} P_0 - 2 \cdot P_1 + P_2 = 0 \\ P_1 - 2 \cdot P_2 + P_3 = 0 \\ \vdots \\ P_{N-2} - 2 \cdot P_{N-1} + P_N = 0 \end{gathered} \tag{5}$$

This set of N-1 equations with N+1 unknown control points, can be developed in matrix form easily (6):

$$\begin{bmatrix} 1 & -2 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow [S] \cdot [P] = [0] \tag{6}$$

A similar approach is used in (7) to minimize the third derivatives considering the third differences of the control polygon.

$$\begin{gathered} -P_0 + 3 \cdot P_1 - 3P_2 + P_3 = 0 \\ -P_1 + 3 \cdot P_2 - 3P_3 + P_4 = 0 \\ \vdots \\ -P_{N-3} + 3 \cdot P_{N-2} - 3P_{N-1} + P_N = 0 \end{gathered} \tag{7}$$

This is expressed in matrix form in (8).

$$\begin{bmatrix} -1 & 3 & -3 & 1 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -3 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 3 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow [T] \cdot [P] = [0] \tag{8}$$

To summarize, matrix formulations (6) and (8) contain an implicit fairing of the control polygon, and can be included in the definition of the LS fitting since these expressions are a function of the control points. Because a weighted combination of the derivatives/differences is used according to (4), a factor $\beta \in [0,1]$ that counts the effect of the third vs. second differences will also be an input to LS fitting. In the same way, a factor $\gamma \in [0,1]$ is used to consider the effect of fairing vs. fitting.

Equations (5) and (7) are now included into the definition of system (1), producing an extended matrix $[M_E]$ of the LS system of (n+2N-2) x (N+1), that is solved as mentioned in section 2.1. This system is sketched in (9)

$$\begin{aligned}&[M_E]\cdot[P]=[Q_E]\Rightarrow[M_E]^T\cdot[M_E]\cdot[P]=[M_E]^T\cdot[Q_E]\Rightarrow\\&\Rightarrow\left\{(1-\gamma)\cdot[M]^T\cdot[M]+\gamma\cdot(1-\beta)\cdot[S]^T\cdot[S]+\gamma\cdot\beta\cdot[T]^T\cdot[T]\right\}\cdot[P]=(1-\gamma)\cdot[M_E]^T\cdot[Q_E]\end{aligned} \tag{9}$$

The extended matrix $[Q_E]$ is the matrix of the n data points, completed with 2N-2 zeros.

2.4 Obtaining the B-spline

Once the data points of interest have been extracted from the scanned ship hull points, the solutions of (9) are the N+1 control points of the B-spline, and the n parameters of the Q_i scattered data points are obtained with the parameterization described in 2.2 and Max/Med distances from the Q_i to the B-Spline are also available. These three steps of the method, LS, distance parameterization, and fairing conditions based on differences, work well together and do not produce a good result for unordered data if they are employed separately, as will be shown in the examples.

The input data will be N, p, and the weights of γ and β for fairing. The use of these weights allows to gain control over the shape of the curve and decide between the adjustment (low γ), and the fairing (high γ). For unordered points, a certain amount of γ is always needed as will be evidenced in the examples. The effect of β will be better understood after the application examples: the third derivatives increase the fairing (high β) but alter the shape of the B-spline more that the second ones (low β).

The method produces better fitting results if the LS is modified to interpolate the end points of the B-Spline. Ordering the data points is not easy but setting the ends as the lowest and highest points (this is the case of ship stations) or a similar binary condition, can be carried out easily. Similar improvements are expected if the tangent values at the ends are included, again including the angles into the definition of the LS system. The values of these angles are not easy to obtain unless a construction drawing is available which, in practice is not always the case.

3. EXAMPLE

In Fig 4, an example of unordered data is presented. In this figure, n=30 data points present an intentioned disorder to show the combined effect of the LS, parameterization and fairing. The poliline in this figure represents the connection sequence between points. A loop is presented on the left part, which will be a shape constraint. For the sake of simplicity, the effect of fairing of the third derivatives/differences is zero (β=0).

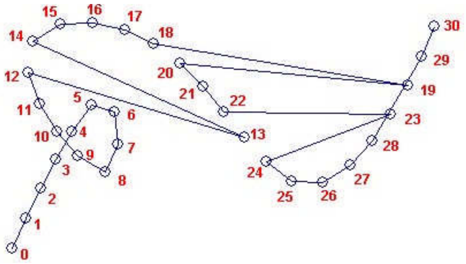


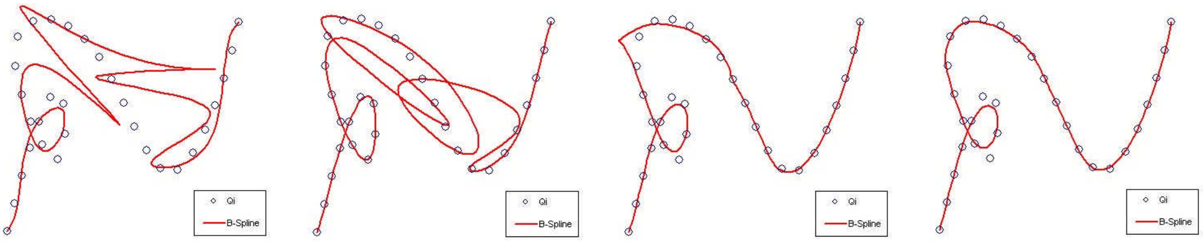
Fig. 4: Example of unordered points.

The Q_i points of Fig. 4 will be approximated with a p=3, N=15 B-spline that interpolates the endpoints and in Fig. 5, different effects of the number of iterations and of the fairing intensity is verified. Referring to Fig. 5, A) shows a direct approach to the problem with the LS of 2.1 and using the centripetal parameterization. A mathematical solution is obtained, but it is clearly not the best fit because of the unordered points. B) depicts an approach considering the iterative parameterization of 2.2 with five iterations (IT=5), but without fairing (γ=0). The aspect of the curve is better. Because the Qi at the loop are ordered, it presents a better fit than A) at this area, but is far from being a good fit: the control polygon wiggles and so does the B-spline. C) presents the first effect of fairing: by maintaining the same number of iterations as B), a small fairing condition of γ=0.005 is imposed to the control polygon. The result is much better than B), but some discontinuities are presented. This is improved in D), by increasing γ=0.01, the discontinuities diminish. Notice the effect on the loop shape.

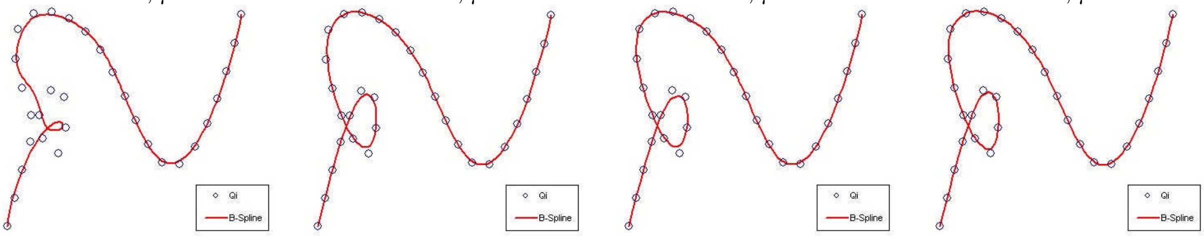
If the fairing intensity increases as in E) γ=0.02, the loop (shape constraint) is not reproduced well. The values of the integrals along the B-Spline, can also be computed to check the numerical fairing criteria of (4): in C) f= 832, in D) f= 653 and in D) f= 549. This experimentally probes that the effect of reducing the differences of the control polygon, reduces the derivatives in the B-spline.

The results of E) can be improved by increasing the number of iterations and maintaining γ. In F) IT=20 and γ is the same as in E). The matching is clearly better at the loop, but at the price of fairing: f = 604, in this case the medium distance d_m from the points to the B-Spline is d_m =0.355 units. Equivalent results can be obtained if IT and γ are both reduced as in G), where d_m =0.417 and f = 577. Finally, the quality of the curve will be a compromise between the different constraints: distance and fairing: H) with IT=30 and γ = 0.01 produces f=602 and d_m = 0.249.

These examples show how the LS, the parameterization, and the fairing based on the differences of the control polygon work well together, and do not produce good results when used separately in unordered points. Different combinations of fairing and iterations produce more or less similar results, but it depends mainly on the number of control points and on the degree of the B-spline. The method is checked in some practical cases as follows.



A: IT=0, γ=0 B: IT=5, γ=0 C: IT=5, γ=0.005 D: IT=5, γ=0.01



E: IT=5, γ=0.02 F: IT=20, γ=0.02 G: IT=10, γ=0.01 H: IT=30, γ=0.01

Fig. 5: Effect of the parameterization and fairing.

4. APPLICATION EXAMPLES

In this section, some stations of the ship's bulbous bow scanned in Fig. 1 are approximated with B-splines according the described method. The stations' points are first extracted from the scanned points of the surface using a Boolean search between two limiting parallel planes separated 2 mm that correspond to the stations of interest, slicing up the surface. In the next figures, half stations are presented for the sake of simplicity since these curves are symmetrical. These half stations are extracted from the stations points, detecting the lowest point, and only considering the points to the left or to the right of the mentioned limit. This technique is used to check the symmetry of the constructed model hull, and the approximated stations can be compared with the design ones of the lines drawing.

Some of the mentioned stations are shown in Fig. 6. This bulbous bow was constructed to match an existing model hull without a bulbous bow, and the matching station within the hull and the bow is depicted if Fig. 6A. This station is measured to check if the bow will match well on the constructed hull. The rest of the stations of Fig. 6 present different sections of the hull approximated with B-splines. Unless another degree is mentioned, the examples show cubic B-Spline and interpolate the lowest and the highest points, which are considered the ends of the B-Spline. These points are also used to check the centerline and the sheerline of the ship/model with the ones of its lines drawing.

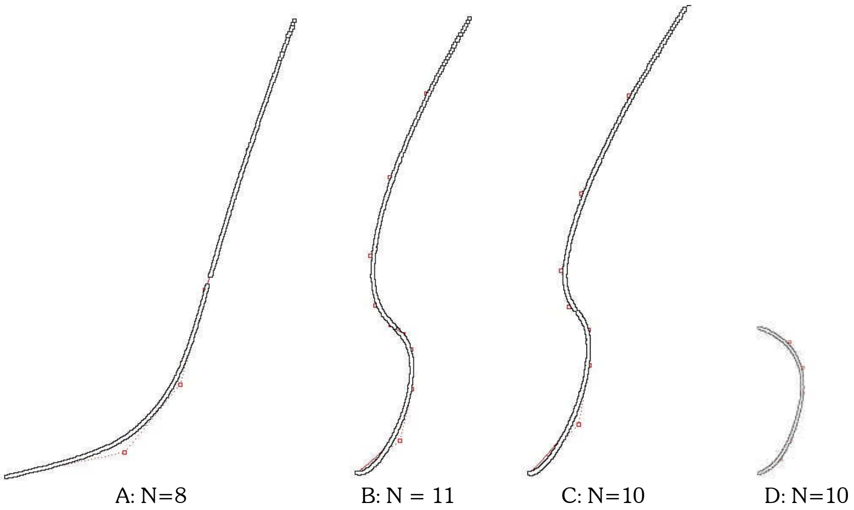


Fig. 6: Stations of the scanned ship's bow.

It is interesting to check the effect of the different components of the method in one of these application examples. Using the same terminology as in section 3, in Fig. 7 A), the data points of the station 6A are approximated with IT = 5 and γ = 0.1 and β = 0. The fit is quite bad. The aspect improves in B), β = 0.5 but is not good enough. The increment of IT improves the fit but again, some wiggles are visually detected in C). This can be solved by increasing the fairing with γ = 0.4 and β = 0 as in D), or increasing the effect of the third derivatives as in E). Both ways are correct, but there is a better tolerance/distance adjustment if β is maintained low.

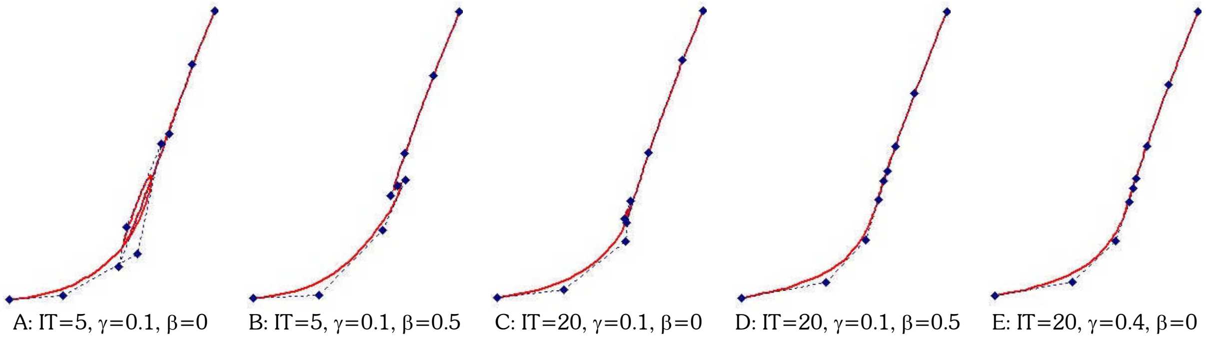


Fig. 7: Analysis of station 6 A.

A similar analysis is made in Fig. 8 of station 6B, which presents an inflection into its definition (x axis scale is magnified to show it better). A low value of γ does not produce a good result. An improvement is made when both IT and γ are increased, until the final result of C). As in the rest of the stations, analogous results can be obtained when considering $\beta>0$, that permits a reduction of γ. As previously mentioned, the stations were fitted with cubic B-Splines, but in Fig. 8 C), D) and E) different degrees are used, maintaining N=11.

Increasing the degree improves the fitting but at the expense of increasing the computational time: p=4 reduces the mean distance of the points to the B-Spline in 13% with respect to the cubic fitting, but the computational time is 7% higher. When p=5 is used, a reduction of 21% percent in distance is achieved, but the computational time rises 68%. When the tolerance fitting is not surpassed, following the principle of the simplest shape, the cubic B-Spline and maintaining N as low as possible are recommended.

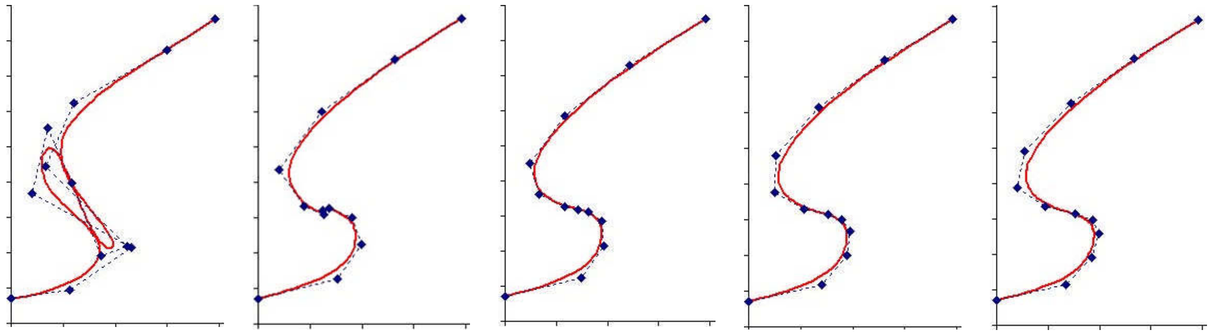


Fig. 8: Analysis of station 6 B.

5. CONCLUSIONS

This paper has presented a method to approximate curves from the data points of a surface obtained from a scanner. The method is suitable when working with unordered data, without a previous ordering, and includes a LS fitting, a distance based parametrization of the data points, and some fairing conditions based on the shape of the control polygon that controls the shape of the B-spline.

The method also allows for different degrees as explained in the application examples and can work with 3D lines. The method considers distance constraints into its definition, so a maximum distance between the data points and the B-Spline is not exceeded, and max/med distances are calculated. The method can reproduce certain shape constraints, e. g. loops as in the example, considering the effect of fairing/fitting, that otherwise could be filtered by a standard approximation.

The method is applied to extract certain lines of a ship hull that present certain characteristics of symmetry, and modify the LS to interpolate certain points of these lines as explained previously in the application examples. The method produces several valid solutions, which are under the prescribed distance tolerance, by considering the effect of fairing/fitting with γ, the issue of the third differences with β and the degree p of the B-Spline. Since all the mentioned solutions are valid, maintaining N and p as low as possible is suggested, following the principle of the simplest shape.

The method can be applied for different applications related with the reconstruction of object lines from scanned data points, and can also be used for ordering scattered data points with the use of the described parameterization. Different applications may use different values of IT, γ and β. The values presented in the application examples work well in the case of this specific application.

6. ACKNOWLEDGEMENTS

The author wish to acknowledge the assistance and comments of Elaine and Pablo

7. REFERENCES

[1] Kobbelt, L.; Botsch, M.: A survey of point-based techniques in computer graphics, Computers and Graphics, 28(6), 2004, 801–14.

[2] Alexa, M.; Behr, J.; Cohen, D.; Fleishman, S.; Levin, D.; Silva, C.: Computing and rendering point set surfaces, IEEE Transactions on Visualizations and Computer Graphics, 9 (1), 2003, 3-15.

[3] Amenta, N.; Bern, J., Kamvysselis, M.: A new Voronoi-based surface reconstruction algorithm, Proceedings of SIGGRAPH'98, 1998, 415-422.

[4] Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W.: Surface reconstruction from unorganized point, Proceedings of SIGGRAPH'92, 1992, 71–8.

[5] Weiss, V.; Andor, L.; Renner, G.; Varady, T.: Advanced surface fitting techniques, Computer-Aided Geometric Design, 19(1), 2002, 19-42.

[6] Yin, Z.: Reverse Engineering of a NURBS surface from digitized points subject to boundary conditions, Computer Graphics, 28(2), 2004, 207-212

[7] Piegl, L. A.; Tiller, W.: The NURBS Book, 2nd Ed. Springer, 1997, 410-413
[8] Jenkins, A.; Traub, J.: A three-stage variable-shift iteration for polynomial zeros and its relation to generalized rayleigh iteration, Numerische Mathematik, 14 (3), 1970, 252-263.
[9] Kjellander, J. A. P.: Smoothing of cubic parametric splints, Computer-Aided Design 15(3), 1983, I75-179.
[10] Farin, G.; Rein, G.; Sapidis, N; Worscy, A.: Faring cubic B-spline curves, Computer-Aided Geometric Design, 4 (1), 1987, 91-104.
[11] Nowacki, H.; Lü, X.: Fairing composite polynomial curves with constraints, Computer-Aided Geometric Design, 11(1), 1994, 1-15.