COMPLEX SURFACE REPRESENTATION

Bruno Crippa* Eva Savina Malinverni* Grazia Tucci*
*D.I.I.A.R. - Politecnico di Milano
P.zza Leonardo da Vinci 32, 20133 Milan (Italy)
E-mail: eva@ipmtf2.topo.polimi.it

ISPRS Commission VI - Working Group 3

Key words: Topological Organization, Convex Hull Reconstruction, Delaunay Triangulation, Bézier Splines

ABSTRACT

In this paper we illustrate a method for automatic reconstruction of models of a piecewise smooth surface. When the object has a complex shape it is very difficult to obtain a good reconstruction and a real-looking representation. If the object has a continuos curvature (convex hull) it is possible to generate a 3D triangular surface employing the *Delaunay* algorithm, generalized to the 3*d*-dimensional case. Then if one wants to have a good result in terms of visual display it is necessary to apply another algorithm that produces *Bézier* splines on a triangular support. Of course, a realistic reconstruction of the object shape needs a high hardware performance to obtain as a final product the display of the interpolated surface (using CAD software). Some applications will be shown by means of examples.

1. Introduction

To create a truly good representation of a three-dimensional object one needs mathematical tools. When the object has a complex shape it is very difficult to obtain its reconstruction, so different methods of interpolation or approximation may be used to generate a Digital Elevation Model (DEM) or Triangulated Irregular Network (TIN), according to the shape characteristics of the object. In fact, if the object has a continuos curvature (convex hull) it is possible to generate a 3D triangular surface employing the *Delaunay* algorithm. Besides, to obtain a good result in terms of visual display it is necessary to apply another algorithm that produces *Bézier* splines on a triangular support. By using this algorithm for the triangulation and smoothing of the generated surface, it is also possible to control the series of points which have to be interpolated and to fix (a priori) the smoothing level of the final surface, with very small fluctuations.

Here a method for automatic reconstruction of models of a piecewise smooth surface is described: some applications will be shown by means of examples.

Of course, a realistic reconstruction of the object shape needs a high hardware performance to obtain as a final product the display of the render of the interpolated surface (using CAD software).

The method presented can be divided into steps:

a) data acquisition;
b) data processing (topological organization);
c) interpolation and new data prediction by *Catmull-Rom* splines;
d) selection of an interpolation technique based on the triangulation method (*Delaunay* algorithm);
e) smoothing of the surface by *Bézier* splines.

2. Data acquisition and topological organization

To obtain a DEM that completely describes an object having a complex surface, it is necessary to survey a lot of points. Data acquisition can be obtained by different and mostly automatic technologies (mechanical coordinate machine taster, laser scanning, photogrammetry, etc..). Among the new techniques, photogrammetry provides the survey of many interesting points with good results. This task unfortunately can't be performed in a short time and without a great effort by human operators using classical phogrammetry (analogical or analytical). On the contrary, digital photogrammetry can solve this problem by applying the fast and automatic method of least squares matching [1] [5] on two or multi images of the object, taken from different points of view. It is important, in this kind of survey, to know not only the coordinates of the points, but also their topology. Data acquisition sometimes results in a series of n points of the object surface in $\Re^3$, which are not always organized in a topological way.

In fact if one takes a series of n points, lying on the external surface of a 3D object, it is important to reconstruct their topology, before applying the algorithms to interpolate or approximate the object surface.

Moreover it is necessary that the whole point set lies on the convex hull, to obtain a digital elevation model close to the object; on the contrary, when there are only few points on the external surface, it obtains a 3D representation is not obtained with an acceptable approximation. Note that there is not only one efficient solution for every possible case.

Data processing, the phase in which the topological structure of the data is fixed, becomes the necessary preliminary step for data reordering, using different criteria.

In this work, the series of given points $I_p=\{P_1, P_2,.., P_n\}$, acquired by photogrammetry, is structured in planar sections (like contour lines or profiles), where the points are classified by different heights (see Fig. 2.1).

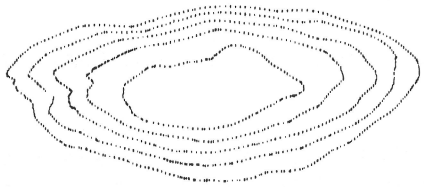


Fig. 2.1 A sample of data in $\Re^3$ structured in contour lines

Topological organization allows for the reordering of the points belonging to the same planar section, researching the relations between the points and then predicting new points.
Scientific literature describes different criteria to this purpose in a short time and with good results [3]. To this aim we decided to apply the *minimum distance criterion*: two points P_i and P_j ,which lie in the same planar section, are neighbouring if and only if they are at the *minimum distance* from one another with respect to any other point of the data set

$$d(P_i, P_j) \le d(P_i, P_k) \quad \text{for every} \quad i \neq j \neq k \qquad (2,1)$$

During this step the new numbering of the points is done too.
The preprocessing is divided into three steps (see Fig.2.2):

1) for every planar sections the coordinates of the centre $C\ (x_c, y_c)$ are calculated:

$$x_c = \frac{\sum_{i=1}^{n} x_i}{n}\ ;\ y_c = \frac{\sum_{i=1}^{n} y_i}{n}\ ;\ z_c = z_i \quad (i = 1, n) \qquad (2,2)$$

2) the reference frame is shifted from the origin x=0 and y=0 to the coordinates of the centre $C\ (x_c, y_c)$:

$$x_i = x_i - x_c \quad ; \quad y_i = y_i - y_c \qquad (2,3)$$

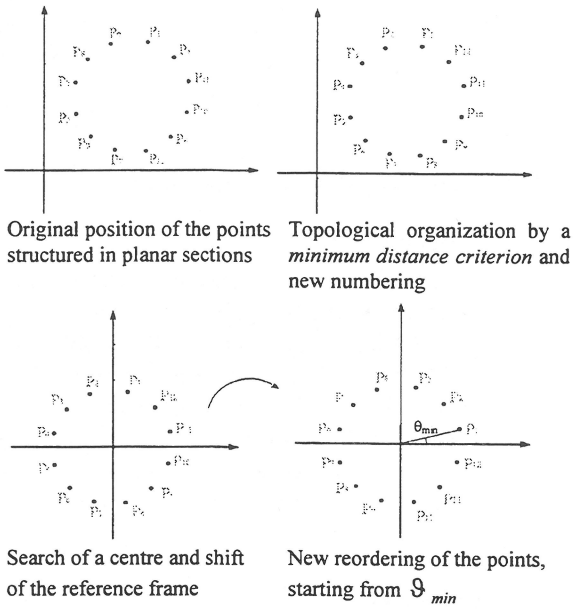
3) for every planar section a new starting point is identified to organize a new numbering using cylindric coordinates with origin in C:

$$P_i(x_i, y_i, z_i) = \{R_i \cos\vartheta_i, R_i \operatorname{sen}\vartheta_i, z_i\} \qquad (2,4)$$

where

$$R_i = \sqrt{(x_i^2 + y_i^2)}$$

$$\vartheta = a\cos\left(\frac{x_i}{R_i}\right) \quad \text{or} \quad \vartheta = a\sin\left(\frac{y_i}{R_i}\right)$$



Original position of the points structured in planar sections

Topological organization by a *minimum distance criterion* and new numbering

Search of a centre and shift of the reference frame

New reordering of the points, starting from ϑ_{min}

Fig. 2.2 Topological organization

The first point is the one with $\vartheta = \vartheta_{min}$, so that the new numbering of the series starts from this point with step ϑ , running counterclockwise.
After data preprocessing it is possible to apply to the reorganized data set interpolating and/or approximating functions which allow for a fast prediction of new data.

3. Interpolation of curves by parametric functions

The classical interpolation problem involves replacing a "complicated" function, *y=f(x)* or *z=f(x,y)*, by a "simpler" function, *y=a(x)* or *z=a(x,y)*, in such a way that the interpolating function and the given function *f* have the same values at positions corresponding to a prescribed set of points.
In this section we present the interpolation of curves given in parametric form (since this is more common in practice), rather than functions of the form *y=f(x)* or *z=f(x,y)*.
The *parametric representation* for curves, *x=f(y)*, *y=f(x)*, *z=z(t)* overcomes the problems caused by functional or implicit forms. Parametric curves replace the use of geometric slopes (which may be infinite) with parametric tangent vectors (which are never infinite).
The image of an open, closed, half open, finite, or infinite interval *I* under a continuos, locally injective mapping into $\Re^2$ or $\Re^3$ is called a curve.
A curve can be considered as a set of points P_i, with respect to a given origin *O*. These points can be regarded as vectors P_i which are the values of a locally one-to-one *vector-valued* function *x=x(t)* of a parameter *t* defined on an interval *I*. The function *x(t)* is called the parametrization of the curve.

Given $n+1$ pairwise distinct points P_i in $\Re^3$, $i=0, 1,...,n$, associated with (appropriately selected) parameters t_i, there are different polynomials which interpolate a curve and different ways to choose the value of the parameter t, depending on the final shape of the interpolating curve, on the computing time, on the accuracy required [4].

We generally prefer the parametrization of curves because it gives us greater flexibility and some advantages. In fact these functions offer:

- less constraints for the control of the shape, since every component (x, y, z) is a function of the parameter t, $x=x(t)$, $y=y(t)$ and $z=(t)$;
- computational advantages and fast programming, since they use the vector-valued form.

The interpolating functions, which are used in particular for many applications in modelling, are polynomials of low degree $3 \le m \le 5$. They describe a given set of empirical data, corresponding to measurements by means of curves with different degrees of smoothness and performed in such a way as to minimize a prescribed error measure and undesirable fluctuations.

So a curve is approximated by a *piecewise polynomial* curve; each segment Q of the overall curve is given by three functions, x, y, and z, which are cubic polynomials in the parameter t.

Cubic polynomials are most often used because lower-degree polynomials give too little flexibility in controlling the shape of the curve, while higher-degree polynomials can introduce unwanted wiggles and also require more computations.

Higher-degree curves require more conditions to determine the coefficients and can "wiggle" back and forth in ways that are difficult to control. Higher-degree curves are used in applications in which higher-degree derivatives must be controlled to create surfaces that are aerodynamically efficient. In fact, the mathematical development for parametric curves and surfaces are often given in terms of an arbitrary degree m. If we fix $m = 3$, the cubic polynomials that define a curve segment $Q(t)=[x(t)\ y(t)\ z(t)]$ are of the form:

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z \qquad 0 \le t \le 1 \end{aligned} \tag{3,1}$$

To deal with finite segments of the curve, without loss of generality, we restrict the parameter t to the $[0,1]$ interval.

Setting $T = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}$, and defining the matrix of the coefficients of the three polynomials as:

$$C = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} \tag{3,2}$$

we can rewrite eq. (3,1) as:

$$Q(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix} = T \cdot C \tag{3,3}$$

This provides a compact way to express eq. (3,1).

If two curve segments are linked together, the curve has G^0 geometric continuity. If the directions (but not necessarily the magnitudes) of the vectors tangent to the two segments are equal at linkage points, the curve has G^1 geometric continuity. In computer-aided design of objects, G^1 continuities between curve segments are equal at the linkage points. If the tangent vectors of two cubic curve segments are equal (i.e., their directions and magnitudes are equal) at the segments' linkage points, the curve has first-degree continuity in the parameter t, or parametric continuity, and is said to be C^1 continuous.

Each cubic polynomial of eq. (3,1) has four coefficients, so four constraints will be needed, allowing us to formulate four equations in the four unknowns, then solving for the unknowns. The three major types of curves are :

- *Hermite* (defined by two endpoints and two endpoint tangent vectors);
- *Bézier* (defined by two endpoints and two more points that control the endpoint tangent vectors);
- *several kinds of splines* (each defined by four control points).

The different types of parametric cubic curves can be compared using different criteria, such as ease of interactive manipulation, degree of continuity at linkage points, generality and computing time needed [4].

To see how the coefficients of eq. (3,1) depend on the four constraints, we rewrite the coefficient matrix as $C = M \cdot G$, where M is a 4x4 *basis matrix*, and G is a four-element column vector of geometric constraints, called the *geometry vector*. The geometric constraints are the conditions, such as endpoints or tangent vectors, that define the curve. The elements of M and G are constants, so the product $T \cdot M \cdot G$ are indeed the *cubic polynomials* in t. Expanding the product $Q(t) = T \cdot M \cdot G$ gives:

$$Q(t) = T \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix} \tag{3,4}$$

Very often, we have a set of positions and want a curve to interpolate (pass through) them smoothly. The *Catmull-Rom* family of interpolating or approximating splines, also called *Overhauser splines*, are useful in this situation. A spline belonging to this family is able to interpolate points P_1 to P_{n-1} in the sequence of points P_0 to P_n. In addition, the tangent vector to point P_i is parallel to the line connecting points P_{i-1} and P_{i+1}. Unfortunately, these splines do not have the convex-hull property. The natural (interpolating) splines also interpolate points, but without the local control guaranteed by the *Catmull-Rom* splines.
Designating M as the *Catmull-Rom* basis matrix and using the same geometry matrix G, the representation is:

$$Q(t) = \frac{1}{2}\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix} \quad (3,5)$$

In this work we have chosen the *Catmull-Rom* cubic spline curves because they have the characteristic of allowing for a fast algorithm [2]. This choice was made also to match these requirements:

1) finding an interpolating smoothing curve with C^1 continuity;
2) finding a curve with only local perturbations, without too many modifications on the complete surface;
3) doing the change of a given point in the series or predicting a new point without having to completely compute the curve again but only the neighbours to the point.

The *Catmull-Rom* cubic spline curves interpolate the point set in the sequence organized by the previous topological criterion. In fact every segment of this curve passes through each point in a parallel direction to the line between the adjacent points with continuos curvature C^1. The straight line segments indicate these directions (see Fig. 3.1).

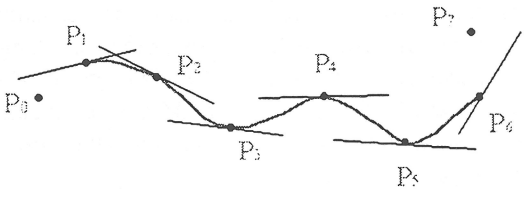


Fig. 3.1 The spline passes through each point in a direction parallel to the line between the adjacent points.

The choice of the parameter is free but depends on the final shape that we want to obtain. In the algorithm implemented and described in this work, we have used an equally spaced parametrization. The prediction of a new point is easy, since we introduce the parametric value t of this new point in the *Catmull-Rom* equation using four points around it. The introduction of a change in the position of a point causes a deviation only in the four segments of curve neighbouring this point. Therefore this type of curve is only locally disturbed. The algorithm implemented uses the series of points previously organized to predict N points on each section, necessary to correctly apply, later on, the *Delaunay* triangulation according to the final complex surface representation. Two examples are presented in Fig.3.2, Fig. 3.3.

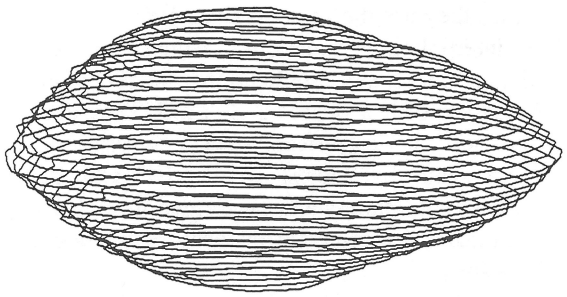


Fig. 3.2 First example of interpolation by *Catmull-Rom* splines

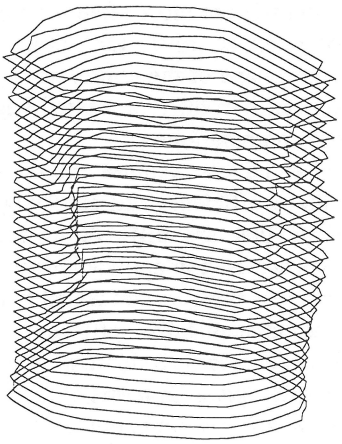


Fig. 3.3 Second example of interpolation by *Catmull-Rom* splines

Note that one must multiply the number N of points by a coefficient k ($k=1$ for curves of odd order and $k=2$ for curves of even order). This procedure will be useful when the *Delaunay* triangulation algorithm will be exploited to research the connections among the points of different planar sections.

4. Application of the *Delaunay* triangulation algorithm

In this section we discuss a method based on the triangulation reconstruction of the convex-hull of the data point set x_i, lying on the planar sections, where the

vertices P_i of the triangulation coincide with the points x_i. For each triangle, we construct a surface patch which interpolates the given function values (and possibly also the derivatives) at the vertices P_i.

There are a number of methods available and they can be differently combined.

The need to construct a globally optimal triangulation suggests that we work with the *d*-dimensional version of the *Delaunay* triangulation (the straight-line dual of the *Voronoi* tessellation). An appropriate triangulation is generally chosen such as to satisfy some optimality criterion which guarantees, first of all, a unique triangulation, possibly without elongated triangles.

A globally optimal triangulation (that is, of course, locally optimal) is the triangulation associated with the *max-min angle criterion*. To explore this, we recall that given a point set $P = \{P_i\}$, the corresponding *Dirichlet* tessellation (see Fig.4.1) (also called the *Thiessen* or *Voronoi* tessellation) is defined as the partition of $\Re^3$ into *Dirichlet* tiles:

$$F_i = \left\{ x \in \Re^2 : d(x, P_i) \le d(x, P_j) \right\} \quad \text{for all } j \neq i \quad (4,1)$$

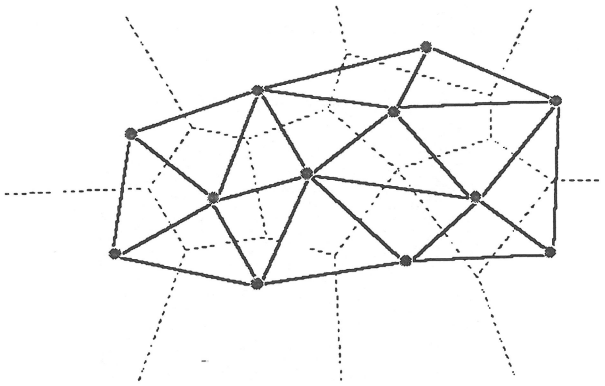


Fig. 4.1 *Voronoi* tessellation and associated *Delaunay* triangulation

Here $d(x, P_k)$ is the *Euclidean* distance, F_i is the polygon consisting of all points $x_i \in \Re^2$ which are closer to P_i than to any other P_j with j≠i. F_i are pairwise disjoint and cover all of $\Re^2$.

Given the point P_i, a corresponding *Dirichlet* tessellation can be constructed by finding the perpendicular bisectors to the line segments connecting the various points P_i.

The *Delaunay* triangulation of the points P_i is the dual of the *Dirichlet* tessellation; two points P_i and P_j are connected if and only if the tiles F_i and F_j of the associated *Dirichlet* Tessellation share a common edge.

The *Delaunay* triangulation can be constructed using an appropriate *"circle criterion"* in 2D dimension or the *"spherical circumscribed circle criterion"* in 3D dimension. For example the local circle criterion is satisfied for a quadrilateral with vertices P_i, P_j, P_k, P_l provided that the circumscribed circle associated with the triangle T_{ijk} with vertices P_i, P_j, P_k does not contain the vertex P_l of the triangle T_{jkl} with vertices P_j, P_k, P_l which shares the edge $P_j P_k$.

If the *local circle criterion* is satisfied for every convex quadrilateral, so is the "strong" *global circle criterion* which requires that for every triangle in the triangulation, the associated circumscribed circle contains no other data point (see Fig.4.2).

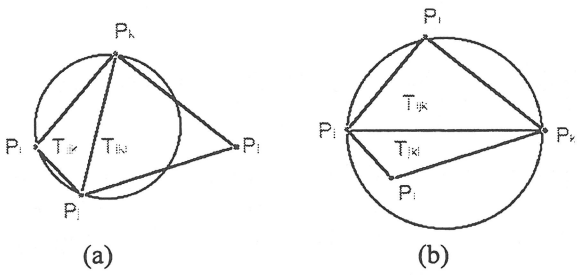


Fig. 4.2 Local circle criterion: (a) satisfied, (b) not satisfied

It is also possible to construct triangulations on curved surfaces using curved triangles (e.g., spherical triangles on the surface of a sphere) (see Fig. 4.3). In fact the *spherical circumscribed circle criterion* checks whether a given point x_l lies inside or outside the spherical circle k_{ijk} passing through the three points x_i, x_j, x_k (see Fig.4.4) [4].

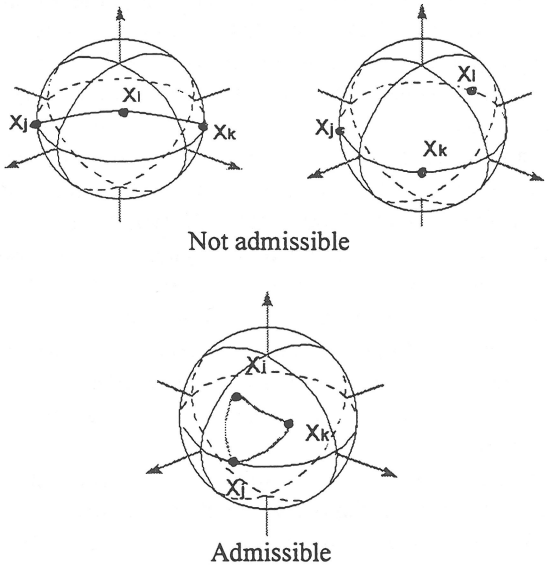


Fig. 4.3 Not admissible and admissible spherical triangles

A triangulation method can be applied to general convex surfaces. Given a set of points $x_i \in \Re^3$, we can define a 3D triangulation, in direct analogy with the definitions reported above also if there are essential

differences between triangulations in the plane and those in $\Re^3$ which must be underlined.

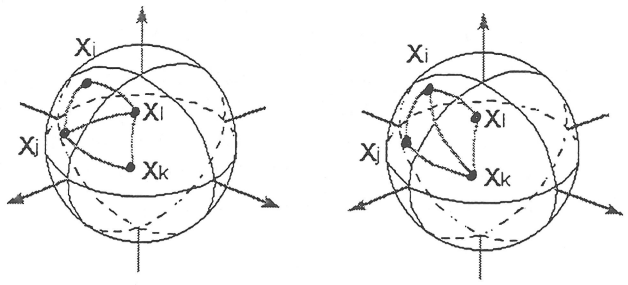


Fig.4.4 Neighbouring triangles with diagonals e_{jl} and e_{ik}

For example in the plane the set of points and its convex-hull uniquely determine the number of triangles and the number of edges of the triangulation. In a higher dimension this is not so simple. So, in $\Re^3$ we cannot always distinguish triangulation on the basis of which data points are connected to each other. Moreover, an iterative construction of the triangulation is not always possible in the three-dimensional case.
While the *max-min angle criterion* can not be directly generalized to $\Re^3$, we can construct the *Delaunay* triangulation by using a version of the *circumscribed circle criterion* involving (hyper) spheres.
We want now to illustrate the method to realize a 3D triangulation of a set of points. In the last section we have shown the topological and interpolating criteria to organize the data. In this phase we search the correspondence among points which belong to planar sections which are next to one another. This is important in order to correctly obtain the final triangulated surface; in fact a wrong correspondence can produce a false interpretation of the model of the complex surface. Among the different methods tested to search corresponding points lying on neighbouring contour lines, we have chosen the *"direction criterion"*. This method is simple and fast also for complex surfaces. It consists in connecting the points, expressed in polar coordinates, of planar sections of different order, which have the same ϑ angle direction.
In this way it is simple to construct the 3D triangulation, because one needs to connect N points predicted on a planar section of odd order with 2N points predicted on a planar section of even order; we recall that the points must have the same ϑ angle direction (see Fig.4.5). Note that in this case we must satisfy the above mentioned *circle-criterion* of *Delaunay*. Two examples are presented in the Fig. 4.6, Fig. 4.7.

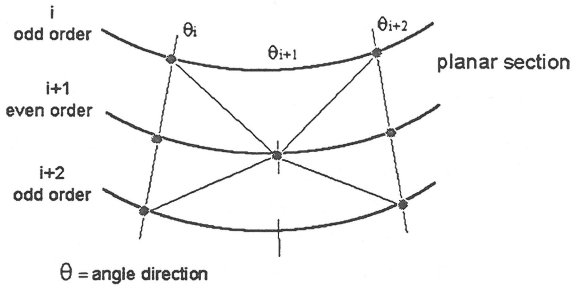


Fig. 4.5 Planar sections of different order: the points with the same angle direction have been connected

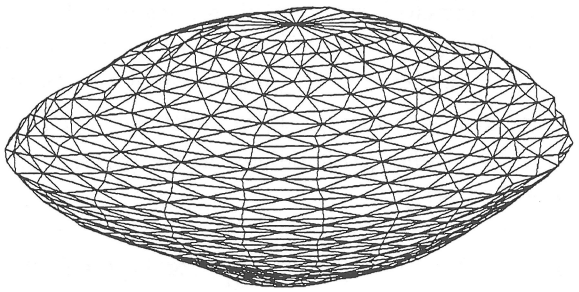


Fig. 4.6 First example of *Delaunay* triangulation

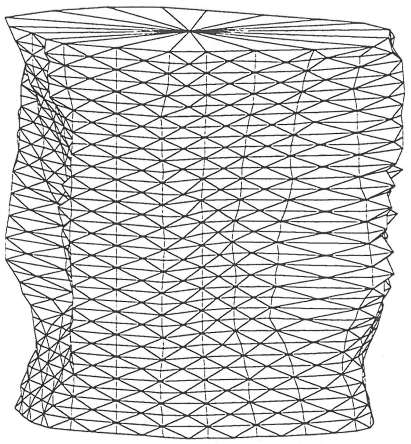


Fig. 4.7 Second example of *Delaunay* triangulation

5. Smoothing of the surface by *Bézier* splines
Now we will describe a type of surface representation by triangular splines on the *Delaunay* triangulation. Afterwards we will show examples of construction and representation of three-dimensional objects closed by a series of surveyed points, obtained fitting the surface constituted by triangular patches.
An interpolating method using global continuous C^r piecewise polynomial functions is defined using the triangular mesh as starting point [4].
The last phase of the surface reconstruction produces a new mesh optimization by exploiting the *Bézier* method. This method, although even if it is locally approximated, allows to obtain good results in terms of visual display. A triangular control mesh is

approximated by a piecewise C^1 continuity spline surface composed by sestic triangular *Bézier* patches.
Modelling of the three-dimensional objects can be obtained through elements (patches) of limited dimension, geometrically simple, easy representable with simple mathematical functions. Every element is formed by many points, whose coordinates are given by continuous parametric functions in two variables (t and s) defined in the limited interval [0,1]. The choice of the type of patch (triangular, quadrilateral, etc.) and the shape of its sides depends on the chosen method of interpolation. For example if the *Delaunay* triangulation is used, every patch is one of the triangles. If, instead, an interpolation with polynomial functions is used, the sides will be curvilinear.
In order to describe such a surface, the parametric equation of the curves (see section 3), extended to the bidimensional case, is:

$$Q(t) = T \cdot M \cdot G \qquad (5,1)$$

and if we decide that the geometric vector G is, instead of constant (as in the case of parametric curves) variable as a function of the parameter t, the parametric surface of the element is obtained. First, for notation convenience, we replace t with s, having $Q(s) = S \cdot M \cdot G$. If we now allow the points in G to vary in 3D along some path which is parameterized on t, we have:

$$Q(s,t) = S \cdot M \cdot \begin{bmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{bmatrix} \text{ with } 0 \le s,t \le 1 \qquad (5,2)$$

Where the geometric vector $G(t)$ becomes a matrix. So $G(t)$ are cubic polynomial functions, and the cubic parametric patches surface is obtained. We have obtained an equation which shows the dependency on s and t, and can again isolate a geometric vector G, which is a constant:

$$Q(s,t) = S \cdot M \cdot G \cdot M^T \cdot T^T \qquad (5,3)$$

The bicubical surface of *Bézier* with regular shape is a surface constituted by rectangular patches, where the geometric matrix G consists of 16 control points (or points of *Bézier*) They are the 16 points that define the polyhedron characteristic, and i.e. the patches of the surface of *Bézier*. They control the slope of the boundary curves and the torsions along the boundary curves.
In many practical applications, when the data are not acquired on a rectangular regular mesh, but they represent a series of scattered points, the choice of a patch of rectangular shape is not convenient since usually triangulation techniques are applied for the construction of the shape. In the examples illustrated in this work the triangular surface patch is considered since it is the more natural choice starting from a triangulation of the points (see Fig. 5.1).

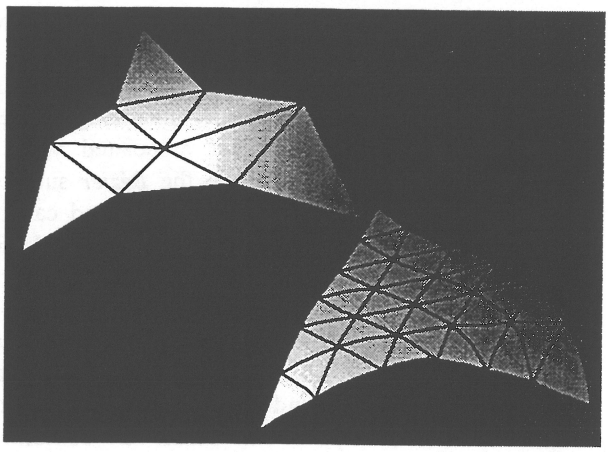


Fig. 5.1 From the triangular mesh to the *Bézier* patches

To analyze this problem in detail, it is convenient to introduce the *Bernstein* polynomials, associated with a base triangle, to construct the parametric equation $X(u)$ of the surface of triangular *Bézier* splines of degree $m=6$. They can be expressed in the following form:

$$X(u) = \sum_{|I|=n} p_I B_I^m(u) \qquad (5,4)$$

where p_I they are the points of *Bézier* and $B_I^m(u)$ is the polynomial of *Bernstein* of degree $m=6$, defined in the following way:

$$B_{|I|}^6(u) = \frac{6!}{i!\,j!\,k!} r^i s^j t^k \qquad (5,5)$$

$i+j+k = 6, \quad i, j, k \ge 0, \quad r+s+t = 1, \quad r, s, t \ge 0,$
$|I| = i+j+k, \quad |u| = r+s+t$

They form the *Bézier* net or *Bézier* polyhedron associated with the surface. It immediately follows that triangular *Bézier* surfaces have the convex-hull property. The convexity of *Bézier* surfaces is not so easy to be determined that of curves. This is a consequence of the fact that a surface with all convex parametric lines is not necessarily convex.
Since for the construction of the spline of *Bézier* the coordinates of the control points are required and only three of them are defined from the initial data (being the vertices of a triangle coming from a previous triangulation of the series of data), the other ones must be obtained using proper criteria, as functions of the coordinates of the vertices of the adjacent triangles to the one taken into account.
An effective method that allows to determine the points of *Bézier* has been proposed by Loop [6]. This method considers the surface approximating the triangles,

whose vertices are the 28 points of *Bézier*. The vertices are computed starting from every side, taking into account the continuity C^1 with the adjacent sides. The method requires to determine four points on every side of the triangular patch and again four points on every segment obtained. Also the direction of the plane of the triangular side to which it belongs is needed.
Therefore the *Bézier* points completely determine the *Bézier* surface, and are also affinity invariantly related to the surface. The *Bézier* net associated with a repeated subdivision converges to the *Bézier* surface. The *Bézier* net approximates the surface, and can be used to compute intersection curves of the *Bézier* surface.
If a single *Bézier* surface is not able to approximate a given surface well enough, then we may use several *Bézier* surface patches which are joined togheter under prescribed continuity conditions (see Fig.5.2). For example we can require that visual C^1 continuity implies geometric C^1 continuity.

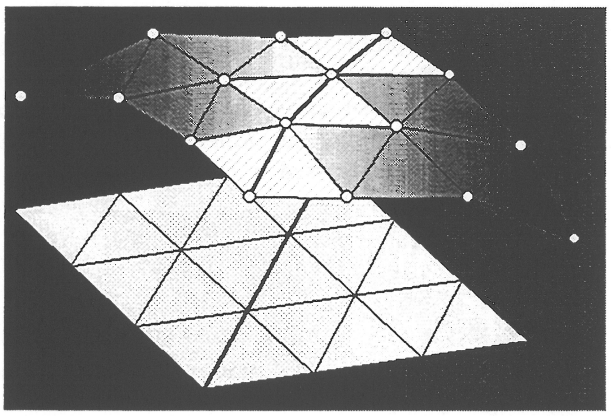


Fig. 5.2 C^1 continuity for the triangular *Bézier* patches

Different cases of continuity must be taken into consideration [4]:

- the first derivatives coincide along and across the common boundary curve between two *Bézier* patches (*C^1 continuity*);
- the first derivatives coincide along the common boundary curve, and the cross derivatives along the boundary curve have the same direction (*visual C^1 continuity*);
- the two neighbouring *Bézier* patches have the same tangent planes along the common boundary curve (*geometric C^1 continuity*)

The parametric continuity C^1 between two patches imposes that the control points of *Bézier* lying on the common side and the neighbouring points are coplanar. In order to guarantee the continuity of the whole surface, besides the continuity conditions between two adjacent elements, it is necessary to guarantee also the continuity condition of all the elements that meet at a vertex. Once solved the continuity conditions of the patch, one has to impose conditions in order to prevent superimpositions of the same patch and to define the normal vector at every point of the common side.
By combining several steps of the algorithm implemented, we can subdivide a triangular *Bézier* patch into an arbitrary number of subtriangles of degree *m*. The sequence of piecewise linear surfaces interpolating the *Bézier* nets converges to the *Bézier* surface (see Fig. 5.3, Fig. 5.4).

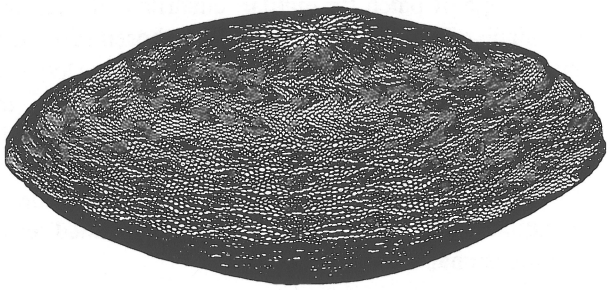


Fig. 5.3 First example of smoothing surface by *Bézier* splines

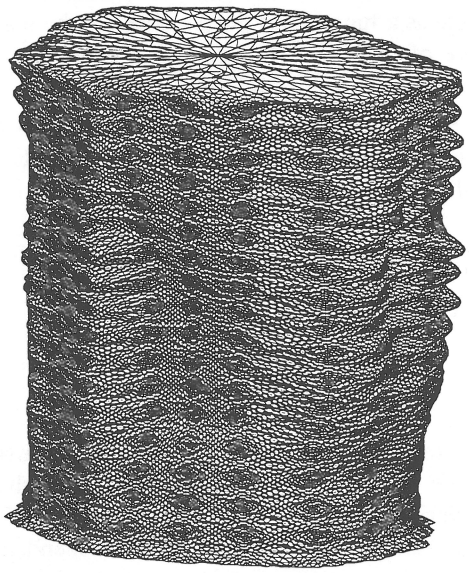


Fig. 5.4 Second example of smoothing surface by *Bézier* splines

6. Conclusion

Usually in designing curves and surfaces, we not only want a good approximation of the data, but we want the curves or surfaces to be *"visually pleasing"*, in some functional or aesthetic way. In the last section of the paper we have described the procedure to generate a good visual final product. Here we present a realistic reconstruction of natural object using CAD software (see Fig. 6.1, Fig. 6.2).

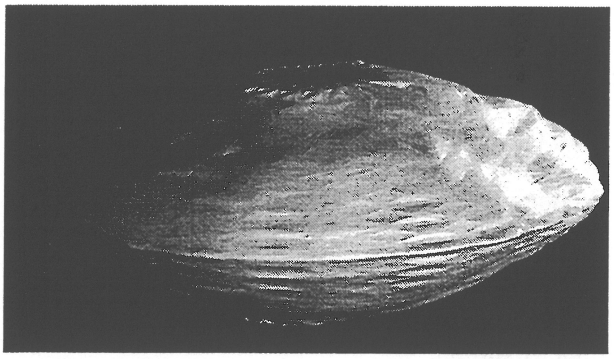


Fig. 6.1 First example: rendering, with a realistic visual display

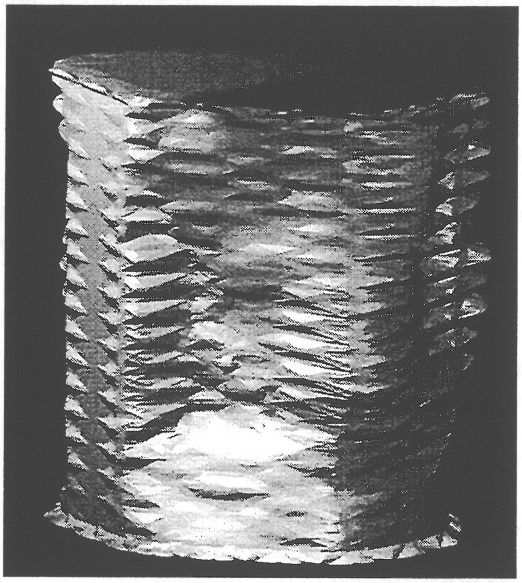


Fig. 6.2 Second example: rendering, with a realistic visual display

As one can see, the technique illustrated gives good results for 3D objects of cylindric or similar shape. The idea of generalizing this method to objects with one or more complex shapes shall be studied, along with the choice of other interpolating functions to predict the points that take into account also the possible roughness of the curves. In conclusion, the problem of the complex surface reconstruction is open to new research in the future.

Acknowledgements
The authors are grateful to Ing. Sharif Omar Sharif, having taken advantages of parts of his PhD Thesis in the field of complex surfaces representation.

References

[1] Baltsavias E.P., 1991: *Multiphoto geometrically constrained matching.* Institut für Geodasie und Photogrammetrie, Mitteilungen n. 49, ETH, Zurich.
[2] Catmull E., Rom R., 1974.: *A Class of Local Interpolating Splines.* in Barnhill, R.E., Riesenfeld R. (ed.), Computer Aided Geometric Disegn. Academic Press, San Francisco.
[3] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., 1990: *Computer Graphics, principles and practice.* Addison-Wesley Publishing Company, Inc.
[4] Hoschek J., Lasser D., 1993: *Fundamentals of Computer Aided Geometric Design.* A. K. Peters, Wellesley, Massachussets.
[5] Kraus K., 1994: *Photogrammetry.* Wichmann Verlag.
[6] Loop C, 1994: *A G^1 triangular spline surface of arbitrary topological type.* Computer Graphics n° 11.