Interpolating Arbitrary Networks of Curves by Catmull-Clark Subdivision Surfaces

Abdulwahed Abbas

The University of Balamand, abbas@balamand.edu.lb

ABSTRACT

This paper pursues an approach for the construction of smooth surfaces for geometric modeling and computer graphics (often referred to as lofting) that starts from a network of polygons representing a set of intersecting curves. The contribution of this paper is in (1) a new method for transforming the initial set of polygons to an equivalent set of polygonal complexes and (2) a new skinning algorithm that adds more vertices and edges thus completing the formation of the control polyhedron. When subdivided, this polyhedron converges on a smooth surface interpolating the initial set of input curves. The resulting surface is C^2, except at points where the initial curves meet, where it is at least C^1. The approach is modular, easy to understand and implement.

Keywords: Cubic B-spline Curves, Doo-Sabin and Catmull-Clark Subdivision Surfaces, Polygonal Complexes, Curve Interpolation.

1. INTRODUCTION

The construction of a smooth surface for computer graphics and computer-aided design applications often start from a sketch. That is, the design often starts from a set of (perhaps intersecting) curves. This is to suggest the constraint that, whatever it is, the surface must pass through these curves.

In this context, the process (often referred to as *lofting*) that starts from the initial curves as input and generates the interpolating surface, with very little (or no) user intervention, is of particular interest to designers [10].

The first problem that such a process should face is that of *compatibility*; that is, not any set of curves can admit a smooth interpolating surface. But, this can always be treated as a background assumption.

The second problem is that a compatible set of curves can admit infinity of interpolating surfaces. For this reason, the process should exploit this degree of freedom by allowing the user to modify the resulting surface to best suit any additional constraints. For many practical reasons, this stage of design is best performed interactively.

In outline, our solution to this problem goes as follows:

1. The curve network is represented by a set of polygons each corresponding to a cubic B-spline curve.
2. A control polyhedron is constructed starting from the curves control points of this network. In other words, this is nothing other than a mesh generation process.
3. When suitably subdivided, this control polyhedron converges on an adequately smooth surface interpolating the initial curve network.

This outline closely parallels that reported in [19]. However, the particular solution of the problem presented in this paper directly follows from our own research on the interpolation problem [15, 16, 17]. It is also less mathematically involved, as it relies on transparent geometric intuition to derive equivalent formulae and transformations, which makes it simpler to understand and to implement.

Since we have essentially covered (in [2] and [3]) the details of how the Catmull-Clark Subdivision algorithm should be adapted to achieve interpolation (item 3 above), the main thrust of this paper is on:

1. showing how the control polyhedron is constructed (item 2 above) and, more importantly perhaps,
2. presenting a single unified framework within which all these tools are integrated.

The input network of intersecting curves defines a corresponding set of patches. Each patch is delimited by a closed loop of curves identifying its boundaries. As such, some patches will appear like wide gaps within the mesh. When a patch is composed of more than four boundary edges, we use a new *skinning* algorithm that adds more control points in its interior, therefore closing the corresponding gap. Finally, a smooth surface is computed using a modification of Catmull-Clark subdivision scheme [7] as specified in [2] and [3].

2. PREVIOUS WORK

Compared with early approaches to the interpolation problem [4, 5, 6], subdivision surfaces provide an altogether simpler framework for handling the problem, especially in computer graphics applications and more so in the context of meshes with arbitrary topology.

The modification of standard Catmull-Clark subdivision coefficients at and around arbitrary intersection points of input curves is necessary, because standard coefficients are incapable of achieving interpolation at extraordinary vertices.

Beside the work of Schaefer et al [19], Levin's *combined* subdivision scheme [11] is an earlier approach to the interpolation problem. This method adjusts the subdivision rules near the curve network to ensure that the surface smoothly interpolates the input curves. Combined subdivision produces surfaces that can interpolate, not just networks of cubic splines, but also arbitrary parametric curves.

Our own work in this area develops subdivision methods for interpolation based on the notion of polygonal complexes [12, 13, 14]. These complexes consist of the portion of a surface mesh defining the curve to be interpolated. As such, it can be integrated as an inherent part of the control mesh. This way, when the mesh is subdivided, the curve will automatically be found lying on the corresponding limit surface.

Polygonal complexes also embody additional information (such as normal and curvature) of the corresponding curve. This allows the designer to adjust the shape of the complex without affecting the interpolation constraints, even though adjustments might affect the nature and quality of the interpolation surface. Therefore, the ability to control adjustments is quite critical for the quality of the overall design [1].

This paper is structured as follows:

- Section 3 reviews relevant subdivision literature; especially with regard to the subdivision of a cubic B-spline curves and also with relevance to both the Doo-Sabin and the Catmull-Clark schemes.
- Section 4 reviews the notion of Polygonal complexes and its relevance as a solution to the interpolation problem.
- Section 5 presents our new skinning algorithm. This algorithm is used to break down a face with more than four edges to faces with at most four edges each. This is done by introducing more vertices and edges within the boundary of that face.
- Section 6 integrates the above intermediate results to show our solution to the interpolation problem.
- Section 7 concludes the paper with some pointers for further research.

3. SUBDIVISION OF POLYGONS AND MESHES

The idea behind subdividing an initial control polygon (resp. mesh) is that more vertices and shorter edges can repeatedly be generate to replace existing ones, up to a limit where this polygon (resp. mesh) converges on a smooth curve (resp. surface).

3.1 The Cubic Subdivision of a Polygon

In a single subdivision step (see Fig. 1), the control polygon $[P_0, P_1, P_2, P_3]$ is subdivided into a polygon $[M_1, N_1, M_2, N_2, M_3]$ as follows:

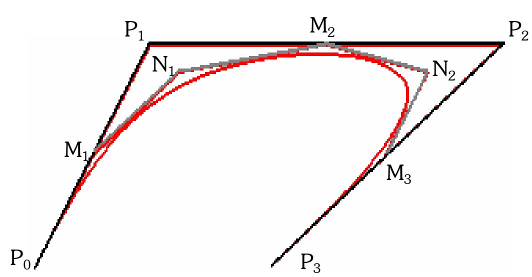

Fig. 1. The Cubic Subdivision of a Polygon.

- M_1 (resp. M_2 and M_3) is the midpoint of the edge P_0P_1 (resp. P_1P_2 and P_2P_3).
- N_1 (resp. N_2) is the midpoint of the edge joining the midpoints of M_1P_1 and P_1M_2 (resp. the midpoint of the edge joining the midpoints of M_2P_2 and P_2M_3).
- Repeating this process sufficiently often leads to a smooth cubic B-spline curve. As a special case, the extremities P_0 and P_3 and of the open polygon can be considered as extremities of the new polygon, and thus be interpolated by the resulting curve.

3.2 The Doo-Sabin Subdivision Scheme

In a single subdivision step [8] (see Fig. 2), for each face *f* of the initial control mesh and for each vertex v_f of this face, a new vertex w_f is computed by suitably averaging the vertices of *f*. The particular coefficients that are used in this calculation are omitted here.

Now, for each face $f = (v_f)$ of the initial control mesh, a new F-face (w_f) is constructed. Similarly, for each inner edge *vv'* joining two faces *f* and *f'*, a new E-face is constructed of the four new vertices w_f, w'_f, $w_{f'}$ and $w'_{f'}$. In the same vain, for each inner vertex v connecting the faces f_i's, a new V-face will be constructed of each *w* vertex calculated from the corresponding *v* vertex with respect to each face f_i.

The subdivided mesh will be obtained from the initial control mesh by connecting these newly composed faces in the obvious way.

Subdividing the initial mesh sufficiently often will result in a smooth limit surface.

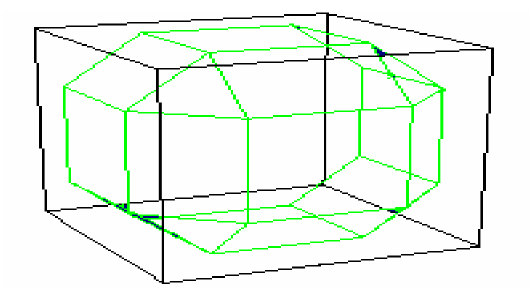

Fig. 2. The Doo-Sabin Subdivision Scheme.

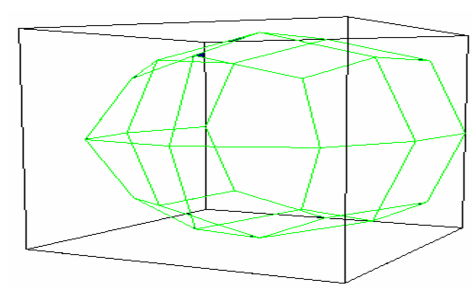

Fig. 3. The Catmull-Clark Subdivision Scheme.

3.3 The Catmull-Clark Subdivision Scheme

In a single subdivision step [7] (see Fig. 3), each face *f* (of the initial control mesh) results in a new F-vertex. Similarly, each edge connecting two faces results in a new E-vertex. In the same vain, each vertex connecting a set of faces results in a new V-vertex. The particular coefficients that are used to calculate the new vertices are omitted here.

Connecting these new vertices in the obvious way will result in a subdivided mesh. Subdividing the initial mesh sufficiently often results in a smooth limit surface.

4. POLYGONAL COMPLEXES FOR INTERPOLATION

A *simple* polygonal complex is a 3×n matrix M of points representing three control polygons top(t_i), middle(m_i) and bottom(b_i), all having the same number n of vertices. These may be seen as a sequence of pairs of rectangular faces where each pair of faces of this sequence has a common edge and each two consecutive pairs have common respective edges (see fig. 4).

A *general* polygonal complex is encountered when the control polygons (t_i), (m_i) and (b_i) do not all have the same number of vertices. That is, the corresponding faces are not all rectangular at the outer edges. However, it is important to note here that each inner vertex of such a complex is regular in the sense that it connects exactly four edges. A general complex leads to a simple one after a single Catmull-Clark subdivision step.

4.1 The Limit of a Polygonal Complex

A Polygonal complex is interesting because, under subdivision, it leads to a sequence of thinner and thinner complexes which, at the limit, converges to a smooth curve.

In this context, the limit of a simple complex M is a B-spline curve whose control polygon is specified by the following formula (see [Nasri&Abbas, 2002]):

$$(1/6)*[1\ 4\ 1]*M \tag{1}$$

Thus, when a complex is embodied within a control mesh, its limit curve is automatically interpolated by the limit surface of this mesh. Likewise, if a complex M' is obtained from a complex M by substituting the mid-polygon m of M by the polygon (see [Nasri et al, 2003]):

$$m' = (1/4)*[-1\ 6\ -1]*M \tag{2}$$

the limit of M' is a B-spline curve identical to that of m. Thus, given a curve defined by a control polygon (m_i)can turn it into a polygonal complex M by adding to it two more rows of points (t_i) and (b_i). The transformation (2), guarantees that any mesh embodying the complex M' interpolates the original curve defined by (m_i).

For the purpose of interpolation, we designate the complex M' as the equivalent of m, since they both generate the same limit curve.

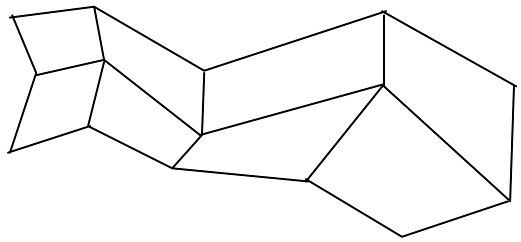

Fig. 4. A Simple Polygonal Complex.

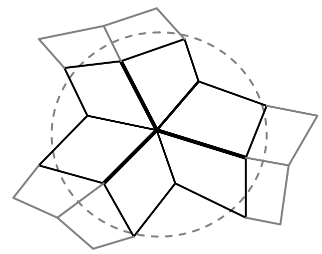

Fig. 5. An X-Configuration.

4.2 X-Configurations and X-Complexes

Now, starting from the initial input set of intersecting polygons, each polygon will be replaced by an equivalent polygonal complex.

Moreover, in the region where the polygons meet, we will arrange it so that the corresponding polygonal complexes meet at so-called X-Configurations [2, 3] (see fig. 5).

Fig. 5 illustrates an X-Configuration where 3 Polygonal complexes meet. The set of Polygonal complexes representing the initial set of polygons is called an X-Complex.

It is obvious now that, by filling in the remaining patches in an X-Complex (see section 5), we will obtain a control mesh. When suitably subdivided, this mesh will yield a smooth surface interpolating the initial input curves.

5. THE SKINNING ROUTINE

The aim here is to fill in a patch delimited by a number of input polygons with additional vertices and edges. This will fill the patch with mostly quadrilateral (and some triangular) faces.

The basic intuition behind this routine should be reminiscent of the routine we devised in [17]. There, the initial input curves for the interpolation task were *parallel* or, in other words, non-intersecting.

Finally, the reader can compare the skinning algorithm below with the quadrangulation algorithm presented in [19].

5.1 The Divide Routine

In the present routine, a face is a closed polygon, which can be considered as an array F of points of length M greater than 2 (see fig. 6).

We need to find the pair on non-adjacent points of indices N and P of this array that are closest to each other in terms of Cartesian distance.

There are many reasons why the shortest distance should be the main criterion for this selection. There are also other criterions that should be satisfied. The most obvious of these is that the edge NP should lie *within the boundaries* of the face, if this face is planer.

Obviously we need to perform this selection only in cases where M is greater than 4. Moreover, in cases where M is greater than 5, the chosen pair of points should be separated, in the array F, with more then a single point.

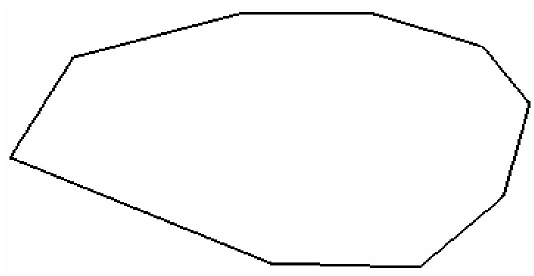

Fig. 6. An Initial Face to be Divided.

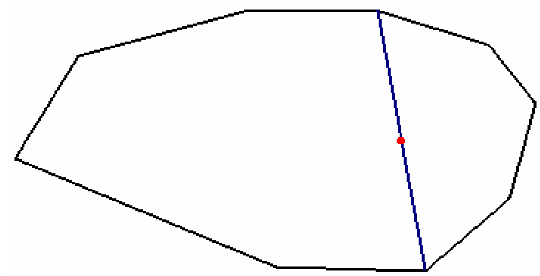

Fig. 7. The Initial Face after a Single Division.

Remembering that *divide* is a basically heuristic routine; we gave it a parameter *d* representing the minimum length of all the edges of the face F. This is used to determine whether the edge NP divides the face F viably with respect to *d*. By viable, we mean here that the distance of all vertices (other than N and P) of the face F to the edge NP is no less than *d*.

Finally, if the edge NP is found to be too long, we subdivide it to sub-edges in such a way that each of these sub-edges is no longer than the maximum length of any of the initial edges of F.

Definition I: a primitive face of the polyhedron is one that consists of no more than four edges.

Definition II: a normal face is one that is convex and where its edges are not too disproportionate in length.

The Divide Routine: this routine splits a non-primitive face f into two viable faces f_1 and f_2. This is done be creating an edge vw of two opposing vertices v and w of the face f. The main criterions for selecting v and w are:

1. the edge vw must lie within the face f.
2. the edge vw is as short as possible. If it turns out that the length of vw is too big with respect to the existing edges of f, it will be subdivided into sub-edges none of which is too big in this respect.
3. the areas of the faces f_1 and f_2 are not too disproportionate.

Clearly, the Divide routine relies on geometric and heuristic intuitions. As such, it will work best on normal faces in the sense of definition II above. However, the shortest criterion guarantees reasonable performance in many awkward situations that might occur in practice.

We mention here that this last part of the routine might leave instances where several of the newly introduced vertices are co-linear. Consequently, fairing techniques [9, 18] might need to be applied to these instances to reduce the negative effect that might have on the resulting surface.

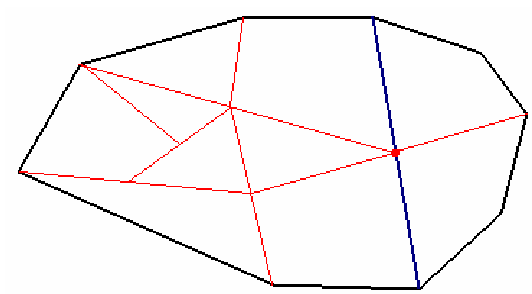

Fig. 8. The Initial Face after Skinning.

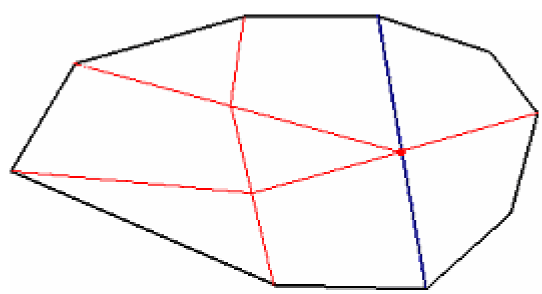

Fig. 9. The Skinned Face after Some Editing (Superfluous Edges Removed).

5.2 The Skin Routine

The divide routine decomposes a face F into two faces F1 and F2, that is. As such, we will need to call this routine repeatedly on the resulting sub-faces until we get at the end a sequence of faces that cannot be decomposed anymore. A face that cannot be decomposed anymore is either a triangle or a quadrilateral (see above definition).

The main skinning algorithm calls the Divide routine on every non-primitive face f of the polyhedron repeatedly. After every call, it replaces f by the corresponding sub-faces f_1 and f_2 until all the faces of the polyhedron are primitive.

Thus, we have, for the overall skin routine, the following loop:

L = a list containing the initial list of faces // *initially*
While L contains a non-primitive face F_0 {
  Remove F_0 from F;
  Divide F_0 into two sub-faces F_1 and F_2;
  Add F_1 and F_2 to L;} // *it is obvious that all the face members of L are now primitive*

The suggested algorithm will introduce more vertices to the initial set of vertices. The overall algorithm will be as good as these new vertices are well-positioned.

6. THE INTERPOLATION PROCESS

The interpolation process can be described as follows:

1. Start with a network of intersecting curves, each is represented by a B-spline polygon (this will define a mesh M_0).
2. Perform an adaptation of the Doo-Sabin F-face F on each of the large faces of M_0. This will give us a polygonal complex Pi containing an initial curves Ci, where Ci is the mid row of Pi. It will also give us a smaller face f lying inside the big face F. Furthermore, this will give us an X-Configuration around each intersection point (see fig. 10). More illustrations are also listed in the appendix.
3. Reposition C_i to (1/4)*[-1 6 -1]*P_i. This will give us a new mesh M_1 from M_0.
4. Perform the skinning routine on each small faces f. This will give us another mesh M_2 from M_1.
5. Iteratively manipulate (fairing, for example) some of the quads on the faces f to improve the positioning of those quads
6. Subdivide according to our modified CC subdivision coefficients

5 This process will give us a surface interpolating the initial set of curves.

6 It is worth noting here that the modified subdivision scheme is employed only at the intersection points. This scheme is explained at length in [3]. Moreover, the resulting surface is C^2, except at the points where the initial curves meet, where it is at least C^1.

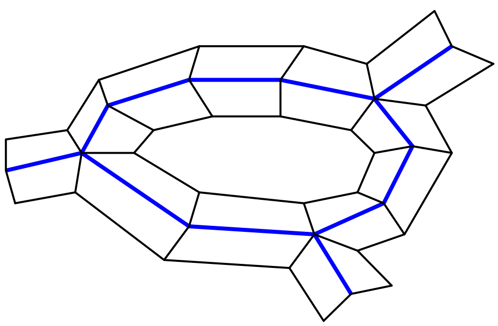

Fig. 10. Doo-Sabin F-Face Application Makes Polygonal Complexes Apparent.

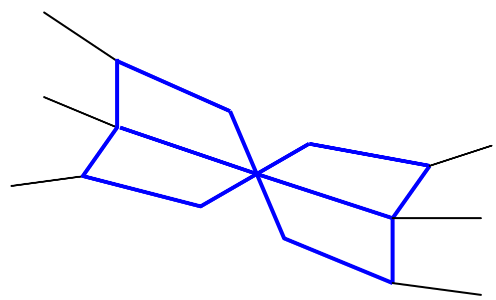

Fig. 11. Symmetric Carry-Through Curve.

This process will give us a surface interpolating the initial set of curves.

It is worth noting here that the modified subdivision scheme is employed only at the intersection points. This scheme is explained at length in [3]. Moreover, the resulting surface is C^2, except at the points where the initial curves meet, where it is at least C^1.

7. CONCLUSIONS AND FURTHER DIRECTIONS

If a curve is to carry through with C^2 continuity then the corresponding polygonal complex should be symmetric near and with respect to the point of intersection. Note here that the point of intersection will be the center of the X-Configuration where the curves passes through.

Note also that, although the polygonal complexes and, consequently, the rest of the X-Complex might seem more awkward to construct; this extra effort will be worthwhile as it will give the designer that much more freedom to tailor the resulting surface to adhere to any further constraints that may be.

Moreover, polygonal complexes have more benefits that are worth mentioning. These are the immediacy, through which the long process seems to follows through, and also the simplicity and transparency of the underlying concepts that have been utilized throughout.

Finally, the generality of the approach is also worth mentioning. In fact, odd or awkward cases seem to be nowhere in sight.

8. ACKNOWLEDGEMENT

This research was supported by a grant from the Lebanese National Council for Scientific Research (LNCSR)

5. REFERENCES

[1] Abbas, A. and Nasri, A., Deforming Subdivision Surfaces with Various Interpolation Constraints, Proceedings of AICCSA' 03, Tunisia, July 2003.

[2] Abbas, A. and Nasri, A., Interpolating Multiple Intersecting Curves Using Catmull-Clark Subdivision Surfaces, *Computer-Aided Design and Applications*, Vol. 1, Nos. 1-4, 2004, pp 25-32.

[3] Abbas, A. and Nasri, A., A Generalized Scheme for the Interpolation Of Arbitrarily Intersecting Curves by Subdivision Surfaces, DEWS, Tokyo, Japan in Feb 24-25, 2005.

[4] Bajaj, C., Lin, K. and Coyle, E., Arbitrary Topology Shape Reconstruction from Planer Cross Sections, *Graphical Models and Image Processing*, Vol. 58, No. 6, 1996, pp 524-543.

[5] Ball, A., Consurf, Part I: Introduction of the Conic Lofting Tile, *Computer Aided Design*, Vol. 6, No. 4, 1994, pp 243-249.

[6] Boissonnat, J., Shape Reconstruction from Planar Cross-Sections, *Computer Vision, Graphics and Image Processing*, Vol. 44, No. 1, 1998, pp 1-29.

[7] Catmull, E. and Clark, J., Recursively Generated B-Spline Surfaces On Arbitrary Meshes, Seminal Graphics, Ed. Rosalee Wolfe, ACM Press, ISBN 1-58113-052-X, 1998, pp 183-188.

[8] Doo, D. and Sabin M., Behavior of Recursive Division Surfaces Near Extraordinary Points, *Computer-Aided Design,* Vol. 10, No. 6, 1978, pp 356-360.

[9] Halstead, M., Kass M. and Derose T., Efficient, Fair Interpolation Using Catmull-Clark Surfaces. SIGGRAPH 93 (August 1993), Computer Graphics Proceedings, Annual Conference Series, 1993, pp 35-44.

[10] Lee, K.-Y., Cho, D.-Y. and Kim, T.W., Interpolation of the Irregular Curve Network of Ship Hullform Using Subdivision Surfaces, *Computer-Aided Design and Applications*, Vol. 1, Nos. 1-4, 2004, pp 17-24.

[11] Levin, A., Interpolating Nets of Curves by Smooth Subdivision Surfaces, In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (1999), ACM Press/Addison-Wesley Publishing Co., 1999, pp 57-64.

[12] Nasri, A., Curve Interpolation in Recursively Generated B-spline Surfaces Over Arbitrary Topology, *Computer Aided Geometric Design*, Vol. 14, No. 1, 1997, pp 13-30.

[13] Nasri, A., Recursive Subdivision of Polygonal Complexes and its Applications in Computer-Aided Geometric Design, *Computer Aided Geometric Design*, Vol. 17, No. 7, 2000, pp 595-619.

[14] Nasri, A., Interpolating an Unlimited Number of Curves Meeting at Extraordinary Points on Subdivision Surfaces, *Computer Graphics Forum,* Vol. 22, No. 1, 2003, pp 87-97.

[15] Nasri, A. and Abbas, A., Lofted Catmull-Clark Subdivision Surfaces, Proceedings of Geometric Modeling and Processing International, 2002, pp. 83-93.

[16] Nasri, A. and Abbas, A., Designing Catmull-Clark Subdivision Surfaces With Curve Interpolation Constraints, *Computers & Graphics*, Vol. 26, No. 3, 2002, pp 393-400.

[17] Nasri, A., Abbas, A. and Hasbini, I., Skinning Catmull-Clark Subdivision Surfaces with Incompatible Cross-Sectional Curves, Proceedings of Pacific Graphics 2003, pp 102-111, Canmore, Canada.

[18] Nasri, A., Kim, T. and Lee, K., Fairing Recursive Subdivision Surfaces with Curve Interpolation Constraints, Proceedings of the International Conference on Shape Modeling and Applications, 2001, pp 49-61.

[19] Schaefer, S., Warren, J. and Zorin, D., Lofting Curve Networks Using Subdivision Surfaces, Eurographics Symposium on Geometric Processing, Scopigno, R. and Zorin, D. (Editors), 2004.

APPENDIX

We list here a sequence of figures that illustrates the working of the process presented in this paper through all its underlying stages.

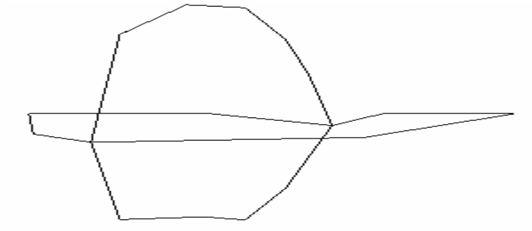

Fig. 12. The initial network of polygons.

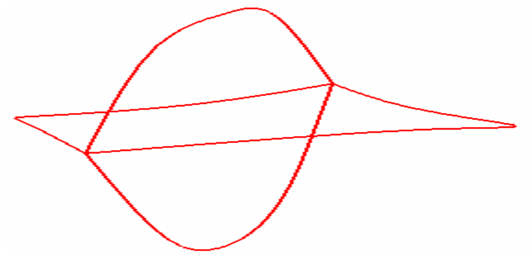

Fig. 13. The Subdivided Cubic B-Spline Curves Corresponding to the above Polygons and Designated for interpolation.

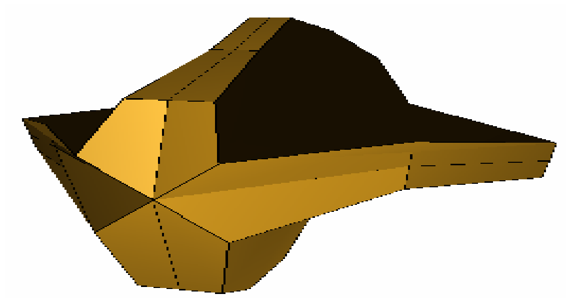

Fig. 14. The Same Network with the Adapted Doo_Sabin Subdivision Creating a Polygonal Complex From Each of the Input Polygons. The Wider Gaps Left on the Mesh are Shown.

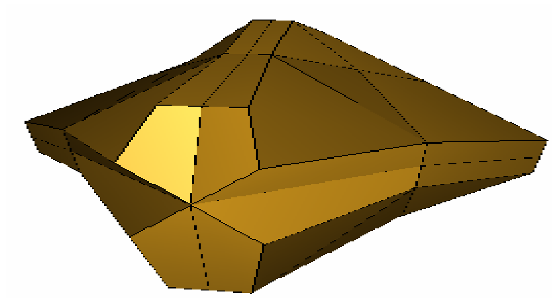

Fig. 15. This illustrates how the Skinning Algorithm Closes the Above-Mentioned Gaps.

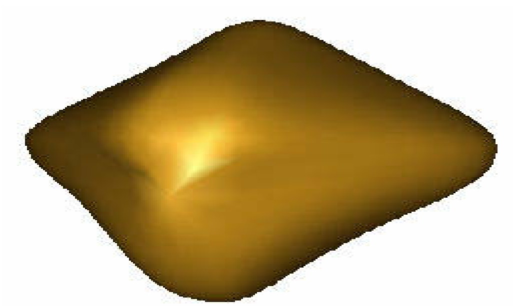

Fig. 16. the Subdivision of the Mesh Into a Smooth Interpolating Surface.

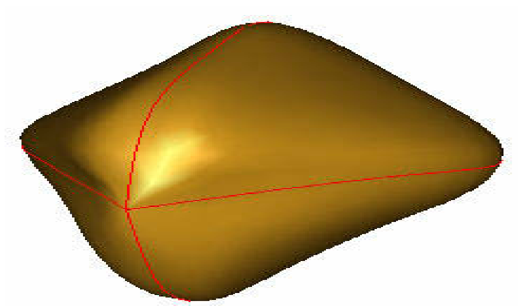

Fig. 17. the Interpolated Curves with one Irregular Intersection Point Clearly Shown.