Survey of Subdivision Controls for Interpolating Curves and Surfaces

Abdulwahed M. Abbas

(Department of Computer Science, The University of Balamand, Tripoli, Lebanon)

Abstract The past few decades witnessed a flurry of research activities in the area of Computer-Aided Geometric Design (CAGD), especially with relation to subdivision surfaces, where particular attention is given to the interpolation of curves or network of curves by such surfaces. This paper traces the major research landmarks pursued in this domain to achieve interpolation of points and curves by such surfaces, specifically by means of polygonal complexes. Along the way, the paper touches upon related research directions, especially the additional interpolation of normal vectors and curvatures following the same approach and under the same settings.

Key words: subdivision surfaces; polygonal complexes; interpolation; normal vectors; curvatures


1 Introduction

Subdivision surfaces\cite{17, 32, 36} are having a penetrating impact across disciplines, from mathematics to computer science and engineering. This is becoming particularly visible in emerging full-length movies\cite{15}, as direct applications of computer graphics and animation techniques\cite{11}. These developments are being witnessed parallel to a flurry of research activities on both the fundamental and the application sides of the field.

The main attraction of recursive subdivision is, first, the unified character of the underlying theory and, second, its flexibility in view of its ability to handle shapes of arbitrary topology. However, the beauty of the underlying theory does not hide shy concerns with relation to:

- Lack of uniform notation and standard analytical tools\cite{35}, especially with regard to the required mathematical foundations, and also to the verification of the smoothness of the limit surface of any arbitrary subdivision scheme\cite{33}. The development of such proofs can sometimes be far from straightforward.

- The suitability of the majority of known (qualitative) subdivision algorithms for (quantitative) engineering designs\cite{37}; e.g., conceptual issues such as continuity and differentiability, accuracy of representation in relation to parameters such...
as normal vectors, curvatures\(^{[2,30]}\) and other practical issues such as the choice between accurate interpolation and approximate fitting\(^{[40]}\).

- Efficiency of subdivision algorithms and their heavy memory consumption in the context of large applications\(^{[31]}\).

Before we continue with this exposition, and in order to clear any possible confusion that may arise, it should perhaps be made clear that this paper is not intended to be a comprehensive survey of recent/current research on recursive subdivision surfaces (for that, the reader can explore existing surveys such as Refs. \([17, 32, 36]\)). This paper is rather an attempt to trace the research done on interpolation during the past decade, especially by means of polygonal complexes\(^{[1,20]}\). The paper goes on to shed some light on the interpolation of other information (beside points and curves) using the same approach and under the same settings.

In achieving curve interpolation by subdivision surfaces, polygonal complexes have the advantage of reducing interpolation, classically-seen as a non-trivial task, to a basic evaluation of a single matrix multiplication. By comparison, this is more immediate and much more efficient than the classical ways of achieving interpolation, which may broadly be classified under the following categories:

- Solving linear systems of equations\(^{[14,34]}\).

- Approximation techniques (usually referred to as fitting), that approaches interpolation down to a given tolerance factor, but perhaps without exactly achieving it\(^{[40]}\).

- Optimization techniques that usually starts from an approximate solution and iteratively working toward an exact solution, to within a given tolerance factor\(^{[13]}\).

The exposition will revolve around the Doo-Sabin\(^{[10]}\) and also the Catmull-Clark\(^{[9]}\) subdivision schemes, mainly because of lack of space and also because most of the applications cited in this paper are implemented on the basis of one or the other of these two schemes.

Within this framework, the comparison with other subdivision schemes, such as Butterfly\(^{[12]}\) or Loop\(^{[16]}\) schemes, may be instructive only when the notion of polygonal complexes are comparatively mature there. We point out here that the development of this notion outside the Doo-Sabin and the Catmull-Clark schemes is just underway\(^{[18, 19, 29]}\).

## 2 Subdivision Surfaces

In the context of a computational process, geometric shapes are more conveniently manipulated through a control mesh, rather than being handled directly. That is, through:

- a control polygon, in the case of curves
- a control polyhedron, in the case of surfaces

For instance, a recursive subdivision surface \(S\) (see Fig. 1) is generally specified by:
A subdivision process repeatedly selects and applies a subdivision rule from the set $R$, taking the initial polyhedron $P$ through a sequence of transformations which, when properly done (see Fig. 2), leads to a smooth limit surface:

![Figure 2. The recursive subdivision process](image)

- Smoothness is crucial for the majority of applications.
- Appropriate methods should be followed for achieving it.

When working toward any specific design goal, most of the effort spent on the production of the limit surface is generally invested on setting the initial control points and later on re-adjusting the positions of those points until the final outcome acceptably matches the expectations of the design. Thus, it would seem reasonable to state that, beside flexibility, a general aim of any modeling scheme would be the reduction of the amount of effort spent during the re-adjustment process.

In this respect, NURBS\cite{34} have long been considered as the standard tool for geometric modeling coming after the initial tensor-product B-spline surfaces. However, recursive subdivision surfaces are now being regarded as an improvement in the sense of freeing the designer from the restriction of regular topology, but perhaps at the cost of diminishing smoothness of the limit surface in the neighborhood of irregular points of the initial control mesh.

Another improvement still came lately by way of T-splines\cite{38}, in the sense of offering, beside few other appealing characteristics, a drastic reduction in the number of initial control points that has to be manipulated at the start of the modeling process.

### 2.1 Doo-Sabin subdivision surfaces

A Doo-Sabin subdivision surface\cite{10} is a generalization of the uniform bi-quadratic B-splines where, starting from an initial control mesh, composed in the usual way of vertices, edges and faces, one subdivision step proceeds by determining (see Fig. 3):

- A new F-face for each face of the mesh.
- A new E-face for each inner edge of the mesh.
A new V-face for each inner vertex of the mesh.

The vertices \( w_i \) of the new faces thus formed are linear combinations of the vertices \( v_i \) of the corresponding faces of the mesh, calculated as indicated in equation (1).

\[
 w_i = \sum_{j=0}^{m} \alpha_{ij} v_j, \quad \text{where} \quad \alpha_{ii} = \frac{n + 5}{4n} \quad \text{and} \quad \alpha_{ij} = \frac{3 + 2\cos(2\pi(i - j)/n)}{4n}
\] (1)

The next mesh in the subdivision sequence is assembled by linking together the various F-, E- and V-faces thus obtained.

2.2 Catmull-Clark subdivision surfaces

A Catmull-Clark subdivision surface\(^9\) is a generalization of the uniform bi-cubic B-spline where, given an initial control mesh, composed of vertices, edges and faces, one subdivision step proceeds by determining:

- A new F-vertex for each face that is the average of the constituent vertices of this face.
- A new E-vertex for each inner edge that is the average of the constituent vertices of this edge together with the F-vertices of the two adjacent old faces.
- A new V-vertex for each inner vertex, determined through the following expression:

\[
\frac{(n - 2)V + R + S}{n}
\]

where

- \( n \) is the valence of the vertex \( V \) of the mesh.
- \( R = \sum_{i=1}^{n} v_i \) and \( S = \sum_{i=1}^{n} v_{f_i} \), where \( V_i \) is a vertex adjacent to \( V \) via an edge and \( V_{f_i} \) is an F-vertex of a face \( f_i \) embodying \( V \).

The next mesh in the subdivision sequence is assembled by linking together the various F-, E- and V-vertices thus obtained (see Fig. 4), in such a way that each E-vertex and V-vertex is connected to the neighboring E-vertices.
2.3 Implementation issues: parallelism and data structure

Due to the enormity of the number of elements (vertices, edges and faces) of the mesh that are generated during the subdivision process, it was inevitable to direct attention to efficiency (in time and in space) of subdivision programming systems. The aim of that is to meaningfully reduce the time taken to generate such elements, as well as the space required to simultaneously store them.

The enhancement that has been attempted in both directions went along designing suitable data structures as well as through parallelizing the corresponding subdivision algorithm so as to benefit from the speed of more than a single processor when subdivision is carried through.

3 Polygonal Complexes

The notion of a polygonal complex is mainly motivated by the need of a polyhedral structure that, under any subdivision scheme, becomes thinner and thinner during the subdivision process till, at the limit, it converges to a curve.\(^\text{[20]}\)

3.1 Doo-Sabin polygonal complexes

Given that motivation, a Doo-Sabin polygonal complex\(^\text{[21]}\) is a sequence of faces \((F_i)_{i}\), where every consecutive pair \((F_i, F_{i+1})\) of the sequence share exactly one edge (see Fig. 5(a)).

Leaving out the outer vertices of the control-polyhedron during subdivision, a polygonal complex becomes narrower and narrower (see Fig. 5(b)) till, at the limit, it becomes a curve (see Fig. 5(c)). This curve is a piecewise quadratic B-spline curve.

The control polygon of this curve is obtained through the following matrix multiplication\(^\text{[1]}\):

\[
\frac{1}{2} \times \begin{bmatrix} 1 & 1 \end{bmatrix} \times M
\]

where, \(M\) is a \(2 \times n\) matrix of the points of the polygonal complex arranged in the obvious way.

It is perhaps worth noting here that the faces of the initial polygonal complex
do not necessarily have to be quads, so long as any two adjacent faces are joined be a single edge (see Fig. 6).

Figure 6. A non-quad-based Doo-Sabin polygonal complex

3.2 Catmull-Clark polygonal complexes

A simple Catmull-Clark polygonal complex\(^{[23]}\) (see Fig. 7(a)) is a \(3 \times n\) matrix \(M\) of vertices representing three control polygons: top\((t_i)\), middle\((m_i)\) and bottom\((b_i)\), all having the same number \(n\) of vertices. By comparison to Doo-Sabin complexes, a Catmull-Clark complex may be seen as a sequence of pairs of rectangular faces; where each pair of the sequence has a common edge and each two consecutive pairs have common respective edges.

Similar to Doo-Sabin complexes and leaving out the outer vertices of the control-polyhedron during subdivision, the polygonal complex gets narrower and narrower (see Fig. 7(b)) during subdivision till, at the limit, it becomes a curve (see Fig. 7(c)).

By contrast with Doo-Sabin complexes, the limit of a simple Catmull-Clark complex \(M\) is a cubic B-spline curve whose control polygon \((P)\) is determined by the following formula\(^{[26]}\):

\[
\frac{1}{6} \times \begin{bmatrix} 1 & 4 & 1 \end{bmatrix} \times M
\]  

(3)

The explicit identity of the limit curve has numerous and interesting applications as will be seen in the following paragraphs.

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, though each inner vertex is regular in the sense that it connects exactly four edges, the corresponding faces do not have to be regular at the outer edges. However, a general complex reduces to a simple one after a single subdivision step (see Fig. 8).

4 Interpolation vs. Approximation

The illustrations of Figs. 5 and 7 make fairly obvious that polygonal complexes should be useful for the interpolation of curves by the limit surface of the corresponding subdivision scheme. In fact, when a complex is embodied within a control mesh, its limit curve is automatically interpolated by the limit surface of the mesh,
quite naturally and without the need for any additional overhead\cite{7}. This applies to Doo-Sabin’s, to Catmull-Clark’s as well as to any other subdivision scheme that the complex may be devised for\cite{7,18}.

This way, there is no more need for extra tools such as solving systems of linear equations\cite{14,34} or for approximation methods\cite{40}, since interpolation is immediate and exact. This approach can be extended to the interpolation of any arbitrary number of isolated curves by a single subdivision surface (see Fig. 9).

4.1 Lofting and/or skinning

In the case of Catmull-Clark polygonal complexes, another perhaps more important benefit is also obtained. In fact, if the mid-polygon $m$ of a simple CC complex $M$ is substituted by the polygon\cite{7,26}:

$$m' = \frac{1}{4} \times [-1 \quad 6 \quad -1] \times M$$

(4)

then the limit curve of the resulting complex $M'$ will be a B-spline curve identical to that of $m$. Accordingly, any given a curve defined by a control polygon ($m_i$), can be turned into a polygonal complex $M$ by adding two more rows of points ($t_i$) and ($b_i$) to it and then, through the application of transformation (4), any mesh embodying the complex $M'$ is in fact guaranteed to be interpolating the original curve defined by ($m_i$).

Similar reverse-transformation may also be devised and applied in the context of Doo-Sabin subdivision surfaces (see Ref. [1] for more details on that).

4.2 Interpolating a sequence of data-points by a cubic B-spline curve

This section gives a brief account of the role of equations (3) and (4) in the interpolation of a sequence of points by a cubic B-spline curve. It should be noted that the results presented in this section are extreme particular cases of those presented in section 3 and earlier on in this section as well.

Given a control polygon (P) having three consecutive points $v_{i-1}$, $v_i$ and $v_{i+1}$,
and a specific point \( v \) that needs to be interpolated by the cubic B-spline curve (C) limit of the control polygon (P), or any variation of that:

- First, with reference to equation (3), the following point is interpolated by the curve (C):
  \[
  \frac{1}{6} \times [1 \ 4 \ 1] \times [v_{i-1} \ v_i \ v_{i+1}]^T
  \]

- Second, with reference to equation (4), when the point \( v_i \) is replaced by the following point:
  \[
  \frac{1}{4} \times [-1 \ 6 \ -1] \times [v_{i-1} \ v \ v_{i+1}]^T
  \]

the curve (\( C' \)), limit of the resulting polygon (\( P' \)) thus obtained, will in fact interpolate \( v \), automatically and without the need for any additional overhead.

Accordingly, given any sequence of points \( v_1, v_2, \ldots, v_n \), a sequence \( p_0, p_1, p_2, \ldots, p_n \) is created in such a way that, for any \( i \), \( v_i \) appears in between \( p_{i-1} \) and \( p_i \). Next, every \( v_i \) in the new sequence is replaced by \( q_i \) where:

\[
q_i = \frac{1}{4} \times [-1 \ 6 \ -1] \times [p_{i-1} \ v_i \ p_{i+1}]^T
\]

Thus, the limit curve of the polygon \( p_0, q_1, p_1, q_2, p_2, \ldots, q_n, p_n \) will interpolate the initial sequence \( (v_i)_i \) as required, without the need for any additional overhead. Figure 10 illustrates two different curves interpolating the same initial sequence of points \( (v_i)_i \).

![Figure 10. Different curves interpolating the same sequence of data points](image)

Obviously, the quality of the resulting curve will depend on the quality of the selection of the auxiliary sequence \( (p_i)_i \). An obvious candidate for the selection method is knot insertion\(^6\).

It is perhaps instructive in this context to point out a comprehensive survey on the kind of constraints that may be imposed on a recursive subdivision curve\(^{24}\) and surface\(^{25}\) in the context of interpolation, as well as all the references that are cited therein.

### 4.3 Interpolating scattered data points

The above techniques and formulae may be used for the generation of a subdivision surface interpolating a collection of scattered data points. However, rather than being arbitrarily scattered, these points are assumed that they can be arranged in a sequence of rows permitting variable number of points in each row\(^4\).

The underlying algorithm runs in two stages. In the first stage, a B-spline curve is generated to interpolate the data points of each row as shown in the previous section, while in the second stage, a subdivision surface is generated to interpolate the curves
outcome of the first stage. As a consequence, this would also interpolate the initial data points, as illustrated in Figs. 11(a), (b) and (c).

![Figure 11. Interpolating scattered data points](image)

### 4.4 Interpolating a network of curves

As opposed to the approach of interpolating individual curves, a possible generalization of that can deal with the situation where two or more such curves are in fact intersecting\(^{[27]}\). In this case, each curve of the network is replaced by an equivalent polygonal complex\(^{[3]}\). The meeting area (called X-Configuration\(^{[5]}\)) of these complexes (see Fig. 12(a)) is arranged in such a way that the meeting point of the intersecting curves is also interpolated by the same subdivision surface (see Fig. 12(b)). Two alternative approaches to the solution are contemplated here:

- either impose extra conditions on the way the initial control points are positioned so as to produce the interpolation effects\(^{[5]}\).
- or change the coefficients of the subdivision scheme being employed so as to produce equivalent effects\(^{[3]}\).

![Figure 12. Interpolating a network of curves](image)

As discussed earlier, the first strategy is generally disliked for the prohibitive effort it consumes to reach the setting required to produce the interpolation effects under the normal subdivision coefficients. The second strategy is thus preferred, though it is naturally associated with having to prove the smoothness of the resulting limit surface. This proof is, however, unconventional and therefore not straightforward to come by, not only because the normal coefficients are modified, but also because smoothness is only exhibited under additional geometric constraints that should be satisfied by the initial control mesh.

It should be noted here that, while the subdivision coefficients are modified around the X-configuration in order to guarantee interpolation, the curves are subdivided normally, thus leaving intact their cubic B-spline identity.

Working without the use of polygonal complexes, an alternative strategy\(^{[39]}\) modifies the subdivision coefficients of both the curves and the surface for the interpolation
effect to take place. However, this approach somewhat obscures the identity of both the interpolated curves and the interpolating surface.

4.5 Filling N-sided polygons

Following the idea of the mesh construction required for the interpolation of a network of curves, and in the case where the polygonal complexes are not trivially short, the resulting mesh will see the appearance of \( n \)-sided faces with fairly large \( n \)'s. The appearance of such faces is generally undesirable because it is associated with the appearance of extra-ordinary vertices during the subdivision process, thus leading to diminishing smoothness of the corresponding limit surface.

It is thus preferable to fill such regions by faces that are regular as far as possible and, therefore, are more suitably handled by subdivision. The process achieving this task\(^{[28]}\) goes through two phases (see Fig. 13, taken from Ref. \([28]\)): a topological phase and a geometrical phase. In the first phase, the connectivity of the mesh is checked for determining a partitioning of the region into sub-regions across which a regular grid could be constructed, while the geometrical phase employs a generalization of discrete Coon’s patches to position the newly-generated vertices in 3D space.

5 Other Applications: control point repositioning

An inner vertex in a Catmull-Clark polygonal complex possesses a four-face neighborhood that may be exploited for holding critical information constraining the nature of the limit surface, not only with relation to the interpolation of the corresponding vertex, but also to the direction of the normal vector and to the value of the curvature of the resulting surface at the vertex being interpolated.

This section provides an additional account of how polygonal complexes may be of use for achieving the interpolation of points, by both curves and by subdivision surfaces. In fact, there are infinitely many ways of achieving interpolation through some simple rules of repositioning of the initial control points. These rules are obtained as multiple solutions of a single equation with many variables.

This should make the method of a very general use, especially for further interpolating tangent/normal vectors as well as prescribed curvatures at such points. The method is also extendible, following the same basic principle to cover the interpolation of a point by Catmull-Clark subdivision surfaces, even under irregular topology.
5.1 Control point repositioning: the cubic curve case

In the general context of interactive modeling or any such setting, when a curve \((C, \text{ limit of a control polygon } (P) = (p_i))\), interpolating a point \(p\), needs to be frequently modified for any reason during the design process, it would be desirable to have available a transition to other alternatives of the infinite number of curves interpolating \(p\) that is smooth and without too much complexity.

Re-iterating what has been said above, in the situation where a control point \(v\) exists between two other control points \(u\) and \(w\), such that:

\[
p = \frac{1}{6} \times [1 \ 4 \ 1] \times [u \ v \ w]^T
\]  

(5)

the control point \(u\) may have to be moved to a new position \(u'\), and so may the points \(v\) (to \(v'\)) and \(w\) (to \(w'\)), for the same limit point \(p\) (or any another designated point \(p'\)) to be interpolated by the cubic B-spline curve \((C')\) limit of the new polygon \((P') = (p'_i)\), obtained as such. These new positions may in fact be better characterized through subtracting equation (6) below from equation (5):

\[
p' = \frac{1}{6} \times [1 \ 4 \ 1] \times [u' \ v' \ w']^T
\]  

(6)

Thus, we get

\[6 \times (p - p') = u - u' + 4 \times (v - v') + w - w'\]  

(7)

In other words,

\[6\delta_p = \delta_u + 4\delta_v + \delta_w\]  

(8)

where \(\delta_p, \delta_u, \delta_v, \) and \(\delta_w\) are vectors representing the amount of shifts that should affect the positions of the points \(p, u, v\) and \(w\) respectively, for the curve \((C')\) to still interpolate the point \(p'\).

The triple-point repositioning, in addition to being simple to obtain, admits many interesting applications, with respect to point as well as to tangent/normal and curvature interpolation. The following sub-sections present illustrative samples of these variations, without attempting to be exhaustive, however.

5.1.1 Point interpolation

This section illustrates the situation were the triple control points \((u, v \text{ and } w)\) may be repositioned in any arbitrary directions with controlled amounts so either to keep the position of the point \(p\) being interpolated intact (i.e. \(\delta_p = 0\)), or to make it move by a prescribed amount in any desirable direction (i.e. \(\delta_p \neq 0\)). The rule here is to simply follow the movement vectors \(\delta_u, \delta_v\) and \(\delta_w\).

* **When** \(\delta_p = 0\)

In the situation where the point \(p\) keeps its position, one of the many ways of satisfying equation (8) is to shift \(u\) (to \(u'\)) and \(w\) (to \(w'\)) twice the distance from \(v\) to \(v'\) but in the opposite direction (see Fig. 14). Thus, a possible solution to equation (8) resides in the following identities: \(\delta_u = \delta_w = -2 \times \delta_v\).

More generally, the above vectors do not have to be parallel in direction, so long as equation (8) holds. For example, if \(p\) holds its position (i.e. \(\delta_p = 0\)), then the
curve \((C')\) will still interpolate the point \(p\) so long as the following constraint holds:

\[
\delta_u + 4\delta_v + \delta_w = 0 \quad \text{see Fig. 15).
\]

* When \(\delta_p \neq 0\)

Alternatively, when the new position of \(p'\) is different from that of \(p\) as a result of such variations, \(p'\) can still be determined in terms of the three movement vectors. For example, if the movement vector at \(v\) reverses direction with double magnitude, \(p'\) will trivially be shifted by the same amount in the same direction (see Fig. 16).

* When \(\delta_u = \delta_w = 0\)

Less trivially perhaps, if \(u\) and \(w\) are to preserve their positions, the amount of shift affecting \(p\) can similarly be determined in terms of the shift affecting \(v\). In fact, from equation (8), we can deduce:

\[
\delta_p = 4 \times \delta_v / 6
\]

This will make it \(2/3\) of the shift occurring to \(v\) and in the same direction (see Fig. 17).

5.1.2 Tangent/Normal interpolation

Since the tangent (T) to the curve \((C)\) at \(p\) is parallel to \(uw\), among the numerous
ways of reaching pre-determined direction of \( T \) (see Fig. 18), an obvious way would be to perform suitable shifts to \( u \) and \( w \) (through rotating the segment \( uw \)).

Other possibilities exist which may involve shifting the position of \( v \) and so perhaps changing the position of \( p \) as well, and again by a prescribed amount.

5.1.3 Curvature interpolation

Without going into too much details, among the numerous ways of reaching a pre-determined modification of the curvature of \( C \) at \( p \) (see Fig. 19), an obvious one would be by performing equal amounts of shifts but in opposite directions to \( u \) and \( w \) (thus modifying the distance from \( u \) to \( w \)). Such a move does not affect the interpolation of \( p \) by \( C' \).

Other possibilities may involve shifting the position of \( v \) and consequently changing the position of \( p \) as well. In addition to that, one can further investigate the possibility of combining the changes in Figs. 18 and 19 to obtain situations where the interpolation of points, as well as prescribed amounts of tangent/normal and curvature are performed simultaneously.
5.2 Control point repositioning: the bi-cubic surface case

In the same context of control point repositioning, but for a bi-cubic surface this time (see Fig. 20), if point \( v_2 \) is moved to a new position \( v_2' \), the new positions of the neighboring points on the regular control mesh \( (p_{ij}) \) may be determined for the new resulting limit surface to still interpolate the same point \( p \) or any other position \( p' \) of \( p \).

![Initial Surface](image1)

![Intended Control Point Movement](image2)

In order to solve this problem, the exercise is repeated starting from the matrix in equation (10). In fact, the bi-cubic limit surface \( S \) whose control mesh is the polygonal complex \( M \):

\[
(M) = \begin{bmatrix}
  u_1 & v_1 & w_1 \\
  u_2 & v_2 & w_2 \\
  u_3 & v_3 & w_3
\end{bmatrix}
\]  

(10)

Will interpolate the cubic B-spline curve \( C \) limit of the control polygon \( Q \) defined by the following matrix multiplication (a particular case of equation (3)):

\[
(Q) = \frac{1}{6} \times \begin{bmatrix} 1 & 4 & 1 \end{bmatrix} \times \begin{bmatrix}
  u_1 & v_1 & w_1 \\
  u_2 & v_2 & w_2 \\
  u_3 & v_3 & w_3
\end{bmatrix}
\]  

(11)

For the sake of simplicity, assuming that \( Q \) consists of the control points \([u \ v \ w]\), a similar expression (to that of the curve case) of the point \( p \) may be obtained, which is:

\[
p = \frac{(u + 4 \times v + w)}{6}
\]

\[
p = \frac{((u_1 + 4 \times u_2 + u_3)/6 + 4 \times (v_1 + 4 \times v_2 + v_3)/6 + (w_1 + 4 \times w_2 + w_3)/6)/6}{6}
\]

\[
p = \frac{(((u_1 + 4 \times u_2 + u_3) + 4 \times (v_1 + 4 \times v_2 + v_3) + (w_1 + 4 \times w_2 + w_3))/36}{36}
\]

\[
p = \frac{((u_1 + u_3 + w_1 + w_3) + 4 \times (u_2 + w_2 + v_1 + v_3) + 16 \times v_2)/36}{36}
\]

This means that, the repositioning of \( p \) may be expressed in terms of the movements of the points of the matrix \( M \) of equation (10), which are:

\[
36 \times \delta_p = \delta_{u_1} + \delta_{u_3} + \delta_{w_1} + \delta_{w_3} + 4 \times (\delta_{u_2} + \delta_{w_2} + \delta_{v_1} + \delta_{v_3}) + 16 \times \delta_{v_2} \quad (12)
\]

The 10 movement variables that may be manipulated in equation (12) simply means that the possibilities though which interpolation can be achieved in this context...
are too numerous to be exhaustively listed in this paper. For this reason, only a few of these cases are discussed, enough to give an idea of how this may be done in practice.

5.2.1. When $\delta_p = 0$

Assuming that $v$ is affected by the same amount of shift as $v_2$ and in the same amount (see Fig. 21), one way of keeping the point $p$ (of Fig. 20) intact, when moving $v_2$ to $v'_2$, could be by shifting the points $u$ and $w$, as depicted in Fig. 14 above, for example.

![Figure 21. The limit surface](image1)

However, since $u$, $v$ and $w$ are themselves limits (by the same operation of equation (5)) of the polygons $[u_1 u_2 u_3]$, $[v_1 v_2 v_3]$ and $[w_1 w_2 w_3]$, respectively, appropriate movements (see Fig. 22) will have to also affect all the remaining points, but this time according to the moves depicted in Fig. 16(a), for example.

Accordingly, the point $p$ interpolated by the initial limit surface will still be interpolated by the resulting limit surface. In other words, similar to shifting of the points of a polygon in the curve case, the faces of a polygonal complex in the case of a surface have to be shifted in both directions of the control mesh. Here again, the size of the shift may accurately be determined. The shift affecting the middle points $v_1$ and $v_3$, for example, should be identical that of the centre point $v_2$, while the shift of the remaining points: $u_1$, $u_2$, $u_3$, $w_1$, $w_2$ and $w_3$ should be twice the size of that and in the opposite direction.

![Figure 22. Movements of the surface patch](image2)

Alternatively, parallel to the line of reasoning followed in section 5.1, the above
result can more directly be reached from equation (12). Other conclusions can similarly be reached. For example, an alternative set of moves to the ones depicted in Fig. 22, but with the same effects may be seen in Fig. 23. The size of the move occurring here to each other point is 4/5 the size of that of \( v_2 \) and in the opposite direction. The effect of that on the point \( p \) remains the same.

\[
\frac{\delta p}{\delta v_2} = \frac{16}{36} \times \frac{4}{9} \text{ in the same direction.}
\]

\[
\text{Figure 23. An alternative set of moves}
\]

5.2.2 When \( \delta_p \neq 0 \)

When \( v_2 \) is moved to \( v'_2 \), with all surrounding points preserving their positions, similar to the curve case, we would then get \( \delta_p = 16 \times \delta v_2 / 36 \). In other words, \( p \) would be shifted 4/9 of the shift of \( v_2 \) and in the same direction (see Fig. 24).

\[
\text{Figure 24. When the limit point moves}
\]

5.2.3 Further remarks

In the surface case, when altering the positions of control points toward interpolating any given point, it is less appealing to simply shift the consecutive points along a certain polygon. Rather, a better approach should aim to shift a whole patch surrounding the point under consideration. The effects of that can again account for the direction of normal and the value of curvature at the point being interpolated.

6 Further Generalizations: Meshes with Irregular Topology

It is generally acknowledged that obtaining results near extraordinary points, parallel to the ones exhibited in the previous sections, is not a trivial task. How-
ever, this section attempts to show that the techniques presented in the previous sections for dealing with cubic curves and bi-cubic surfaces can directly be generalized to cover, for example, Catmull-Clark subdivision surfaces with meshes exhibiting irregular topology.

In order to reach this goal, it should be sufficient to show that the limit point of an irregular patch is an explicit linear combination of all the points involved in the patch. Subsequently, the treatment employed in the regular case can similarly and naturally be extended to cover irregular cases.

The limit of an inner vertex, of valence $n$, on the initial control mesh $(M)$ is a vertex on the limit surface $(S)$ of $(M)$ given by the following expression:

$$v^\infty = \frac{n^2v^1 + 4\sum_{i=1}^{n}w_i + \sum_{i=1}^{n}f_i}{n(n+5)}$$

where $v_1$, $e_1$ and $f_1$ are respectively the $1^{st}$ level V-vertex, E-vertex and F-vertex corresponding to vertex $v$, edge $e$ and face $f$ having $v$ as a component vertex.

In order to somewhat simplify the presentation, we adopt the convention that a vertex is called $w_i$ when it is linked directly to the vertex $v$ via an edge; otherwise, it is simply named $v_i$. Accordingly:

$$f_i^1 = \frac{v + w_i + \sum_{j=1}^{n-1} v_j + w_{i+1}}{n_i}, e_i^1 = \frac{v + w_i + f_{i-1} + f_i^1}{4}$$

and

$$v^1 = \frac{n(n-2)v + \sum_{i=1}^{n}w_i + \sum_{i=1}^{n}f_i^1}{n^2}$$

which means:

$$v^\infty = \frac{n(n-2)v + \sum_{i=1}^{n}w_i + \sum_{i=1}^{n}f_i^1}{n(n+5)}$$

In other words:

$$v^\infty = \frac{n(n-1)v + 2\sum_{i=1}^{n}w_i + 4\sum_{i=1}^{n}f_i^1}{n(n+5)}$$

Now, replacing the F-vertices by their values, we get:

$$v^\infty = \frac{n(n-1)v + 2\sum_{i=1}^{n}w_i + \sum_{i=1}^{n}f_{i-1} + f_i^1 + \sum_{i=1}^{n}f_i^1}{n(n+5)}$$

Thus, we get:

$$v^\infty = \frac{n(n-1)v + 2\sum_{i=1}^{n}w_i + \sum_{i=1}^{n}f_{i-1} + f_i^1 + \sum_{i=1}^{n}f_i^1}{n(n+5)}$$

For verification purposes, in the regular case, when $n$ and all other $n_i$s are uniformly equal to 4, the previous expression becomes:

$$v^\infty = \frac{16v + 4\sum_{i=1}^{n}w_i + \sum_{i=1}^{n}w_i}{36}$$

This exactly mirrors the expression of the limit vertex described in section 5.2 for the regular case.

7 Conclusions and Further Research

The previous paragraphs provide a review of the approach to interpolation by subdivision surfaces, mainly by means of polygonal complexes. The discussion revolves more around Catmull-Clark subdivision surfaces and especially around the manipulation cubic uniform B-spline curves as an extreme particular application of that. Through cubic B-spline curves, in addition to the interpolation of points, the paper also shed some light on the interpolation of normal vectors and curvatures at the points being interpolated in the context of bi-cubic spline surfaces, as well as in the context of Catmull-Clark subdivision surfaces, with meshes of arbitrary topology.
But for lack of space, we could have followed this track by providing more material on the interpolation of curves by subdivision surfaces. However, this track may be followed using the pointers and the references included in this paper.

Due their localized effects, the paper clearly recommends that the control-point movements are better used in an interactive setting, because of the artifacts that are otherwise likely to appear on the resulting curve or surface. In this respect, more research work is required to safeguard their application in stand-alone modeling systems. At present, most interpolation techniques cited in the literature invariably call for fairing\cite{22} and fairing techniques to minimize the impact of such undesirable side-effects.

Another plausible application of the control-point movements is in the context of iterative approximation technique\cite{13}. Very briefly, this is a less-efficient technique that is of current research interest which, for lack of a better way, repeatedly modifies an initial shape towards a final state satisfying the desirable constraints.

In the same vein, we point to another issue that has not been addressed in this paper. This is the generation of an interpolating curve or a surface satisfying uniform normal vector variation as well as uniform curvature distribution all over the interpolating dimension. The subtlety of this problem, and the absence of more direct techniques to solve it at the present time, led researchers to again adopt the same iterative approximation treatment\cite{13,30} that can approach the interpolation curve or a surface down to any desirable level of tolerance.

Finally, in addition to the interpolation of points and curves, a more interesting track to pursue is perhaps to generalize the interpolation of normal vectors and curvatures from curves to surfaces. This would be a considerable step to take especially with respect to the interpolating of arbitrarily intersecting curves by subdivision surfaces, for the huge enhancement this will have on the expressive power of subdivision surfaces in general.

**Acknowledgement**

Thanks are due to Ahmad Nasri for his advice and support during the development of the ideas and of the initial manuscript leading to this paper. Thanks are also due to the anonymous reviewers whose constructive comments help in enormously improving the quality of the presented material.

**References**


Abdulwahed M. Abbas: Survey of subdivision controls for interpolating ...


