SUBDIVISION OF CURVES AND SURFACES: AN OVERVIEW

Ben Herbst, Karin M Hunter, Emile Rossouw

Applied Mathematics, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa

Abstract: Subdivision schemes are widely used in various applications such as data-fitting, computer graphics and solid modeling. In this paper we present the basic ideas of subdivision schemes for curves; both interpolatory and corner-cutting schemes, as well as their adaptation to finite sequences. We conclude with examples of specific applications for these subdivision schemes and provide an example of surface subdivision.

Key Words: subdivision, curve fitting, approximation theory

1. INTRODUCTION

Few people fail to be impressed by the quality of the graphics of recent animated movies. Geri's Game by Pixar is a beautiful example. The keen observer will note the names of Edwin Catmull and Jim Clark among the credentials. Their contributions to animation, based on their subdivision scheme for surfaces, won them an Academy Award for Technical Achievement in 2006.

To understand what subdivision is all about, one should realise that the quality of the three-dimensional graphics depend, among others, on the modeling of the objects themselves. As in the case of Geri, one wants to construct a face based on a limited number of control points that defines the basic structure. This implies that given the control points, the region between the control points should be constructed in a realistic way. One can of course use interpolation, typically spline interpolation in which case the interpolant is first constructed and then evaluated at the required points. Should one decide to move one of the control points the process starts all over. Subdivision schemes skip the first step of constructing the interpolant. Instead it proceeds directly from the control points to the filled-out surface through an iterative procedure. Moreover, the process is local with the advantage that any change in control points have only a local effect. Changing Geri's nose by moving a control point does not for example, affect his mouth. Apart from computer animation, subdivision schemes also find wide applications in computer graphics and solid modeling.

In this paper we explain the basic ideas behind subdivision schemes on curves before we briefly indicate how these ideas carry over to the subdivision of surfaces.

Suppose we are given a bi-infinite sequence of points $c^{(0)} = \{c_j^{(0)} : j \in \mathbb{Z}\}$, and are interested in approximating these points with a smooth curve. Standard ways of doing this involves constructing an approximating function, e.g. an interpolating spline. Then, in order to visually represent the approximating function in computer applications, the function needs to be evaluated on a sufficiently dense set of points. Subdivision schemes skip the first step by creating a dense set of points directly from the given points, i.e. there is no need to first construct the approximating function and then evaluate it. This leads to considerable savings in computational cost.

Of course, if the original points contain noise, the approximating curve should not pass through them, i.e. the approximating curve should not be interpolatory. This can be achieved in different ways: It is possible to first apply a smoothing operation to the given points and then do an interpolation, or one can use something like a smoothing spline. In this paper we describe two techniques of subdivision, one interpolatory and one smoothing (or corner-cutting).

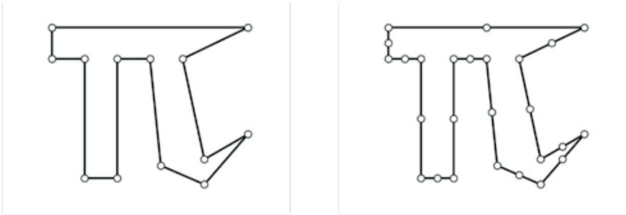
Consider the following simple iterative procedure: Start with a set of points $c^{(0)}$, called the *original control points*, and generate a new set of control points $c^{(1)} = \{c_j^{(1)} : j \in \mathbb{Z}\}$ by taking a linear combination of the original control points. Repeat this until the desired density is achieved.

For example if one generates the new control points using the simple linear combination

$$c_{2j}^{(1)} = c_j^{(0)} \text{ and } c_{2j+1}^{(1)} = \tfrac{1}{2}(c_j^{(0)} + c_{j+1}^{(0)}), \quad j \in \mathbb{Z}, \quad (1)$$

then the even-indexed elements of the new control points are simply the original points and the odd-indexed elements are generated halfway between the old control points. This step is then repeated indefinitely, roughly doubling the number of points at each step. In this case the points fill in or converge to the straight-line segments connecting the original control points, as illustrated in Figure 1. Thus we obtain a continuous piecewise linear curve. This simple procedure is an example of a subdivision scheme.

In general, given a sequence $a = \{a_j : j \in \mathbb{Z}\}$, called



(a) original control points *(b) control points after one iteration*

Figure 1: Illustration of the iterative procedure Equation 1.

the *mask* of the subdivision scheme, and a sequence of control points $c = \{c_j : j \in \mathbb{Z}\}$, we define the corresponding subdivision operator S_a by

$$(S_a c)_j = \sum_{k \in \mathbb{Z}} a_{j-2k} c_k, \quad j \in \mathbb{Z}. \tag{2}$$

The resulting subdivision scheme is then defined, for a given sequence of control points c, by

$$c^{(0)} = c, \quad c^{(r+1)} = S_a c^{(r)}, \quad r = 0, 1, \ldots, \tag{3}$$

or, equivalently,

$$c^{(0)} = c, \quad c^{(r+1)} = S_a^{r+1} c, \quad r = 0, 1, \ldots. \tag{4}$$

Note that the nature of the subdivision scheme is entirely determined by the choice of the mask, i.e. the linear combination used to generate the new control points at each iteration. The key therefore is to find an appropriate mask. The choice of the mask determines (i) whether the subdivision scheme is interpolatory or smoothing (corner-cutting), (ii) the convergence of the scheme, and (iii) the smoothness of the final curve.

These issues are closely related to the existence of a *refinable function*, as briefly discussed in the sections below, and refinable functions provide a link to wavelets, see e.g. [1].

There is no unique or best way of choosing a mask. One possibility is to demand that if the original control points fall on a polynomial of a certain degree, then the newly generated control points must lie on the same polynomial. This is, in fact, the idea behind Dubuc-Deslauriers subdivision scheme [2, 3]. In Section 2, we develop this idea to derive explicit formulae for the Dubuc-Deslauriers mask and provide an adaptation for finite control sequences.

In Section 3 we introduce corner-cutting subdivision schemes and their corresponding refinable functions, for infinite as well as finite sequences of control points.

A few applications of the subdivision of curves are given in Section 4 and surface subdivision is briefly discussed in Section 5.

2. INTERPOLATORY SUBDIVISION

In this section we introduce the well known Dubuc-Deslauriers subdivision scheme as an optimally local, curve filling iterative procedure that reproduces polynomials of a given odd degree. We then indicate how the limit curve for the Dubuc-Deslauriers scheme depends on the existence of an associated refinable function and provide an adaption of this scheme for finite sequences.

2.1. *Construction of the mask*

Suppose the original control points fall on a polynomial p of degree $2n-1$, i.e. $c_j^{(0)} = p(j)$, $j \in \mathbb{Z}$. Consider the problem of finding the shortest possible mask a such that all the subsequent iteratives $c^{(r)}$ fall on the same polynomial. More specifically, we require that

$$\sum_{k \in \mathbb{Z}} a_{j-2k} p(k) = p\left(\frac{j}{2}\right), \quad j \in \mathbb{Z}. \tag{5}$$

The mask is derived from the standard Lagrange polynomials of degree $2n-1$, uniquely defined by

$$\ell_k(j) = \delta_{k,j}, \quad k, j = -n+1, \ldots, n, \tag{6}$$

and therefore satisfying the polynomial reproduction property

$$\sum_{k=-n+1}^{n} p(k)\ell_k(x) = p(x), \quad x \in \mathbb{R}. \tag{7}$$

An explicit formula for these Lagrange functions, for $k = -n+1, \ldots, n$, is given by

$$\ell_k(x) = \prod_{\substack{k=-n+1 \\ k \neq j}}^{n} \frac{x-j}{k-j}, \quad x \in \mathbb{R}. \tag{8}$$

Comparing Equation 7 (with $x = \frac{1}{2}$) and Equation 5 (with $j = 1$), it follows readily that the shortest possible mask satisfying Equation 5 is given by

$$\begin{aligned} a_{2j} &= \delta_{j,0}, \qquad j \in \mathbb{Z} && \text{(9a)} \\ a_{2j+1} &= \ell_{-j}\left(\tfrac{1}{2}\right), \ j = -n \ldots, n-1, && \text{(9b)} \\ a_{2j+1} &= 0, \qquad \text{otherwise.} && \text{(9c)} \end{aligned}$$

This mask is known as the Dubuc-Deslauriers mask [3].

Note that Equation 9a implies that Equation 3 satisfies the interpolatory property

$$c_{2j}^{(r+1)} = c_j^{(r)}, \quad j \in \mathbb{Z}, \ r = 0, 1, \ldots, \tag{10}$$

i.e. the Dubuc-Deslauriers subdivision scheme is *interpolatory*. Also, the mask coefficients are symmetric, i.e.

$$a_j = a_{-j}, \quad j \in \mathbb{Z}. \tag{11}$$

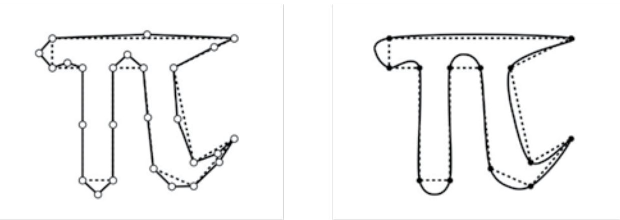
Therefore the Dubuc-Deslauriers subdivision scheme is interpolatory, symmetric and fills polynomials of degree $2n-1$.

For example, if $n=1$, Equation 9, Equation 2 and Equation 3 yield for $r=0$, the iteration procedure of Equation 1. This subdivision scheme converges to a continuous piecewise linear function that interpolates the original control points (see Figure 1).

For $n=2$ we get

$$a_{2j+1} = \begin{cases} -\frac{1}{16}, & j=-2, \\ \frac{9}{16}, & j=-1, \\ \frac{9}{16}, & j=0, \\ -\frac{1}{16}, & j=1, \\ 0, & \text{otherwise.} \end{cases} \qquad (12)$$

Subdivision with this mask converges to a smooth function [3], while still interpolating the original control points, see Figure 2.



(a) control points: original (•), after one iteration (∘)   *(b) original control points (•), control polygon after 6 iterations (−)*

Figure 2: Dubuc-Deslauriers subdivision for $n=2$

It is interesting to note that Knuth based his construction of TEX fonts [4, Chapter 2] on ideas remarkably similar to subdivision schemes more than 10 years before the Dubuc-Deslauriers scheme was introduced in [3].

Figure 2 suggests that the Dubuc-Deslauriers subdivision converges to a smooth function for $n=2$. For a proof see [3, 5]. This limiting curve is described in terms of a *refinable* function, to be discussed in the next section.

2.2. *Convergence of Dubuc-Deslauriers subdivision*

The mask a of a convergent subdivision scheme ensures the existence of a function ϕ satisfying

$$\phi(x) = \sum_{j\in\mathbb{Z}} a_j \phi(2x-j), \quad x\in\mathbb{R}. \qquad (13)$$

We call such a function an *refinable function.*

It is shown in [3, 5] that the Dubuc-Deslauriers mask a generates a convergent subdivision scheme, which then guarantees the existence of an associated refinable function ϕ. Moreover, the refinable function inherits the mask's finite support and symmetry, as well as its polynomial filling and interpolatory properties as follows

$$\phi(x) = 0, \qquad x \notin (-2n+1, 2n-1), \quad (14)$$
$$\phi(x) = \phi(-x), \qquad x\in\mathbb{R}, \quad (15)$$
$$\sum_{j\in\mathbb{Z}} p(j)\phi(x-j) = p(x), \quad x\in\mathbb{R}, \quad (16)$$
$$\phi(j) = \delta_{j,0}, \qquad j\in\mathbb{Z}, \quad (17)$$

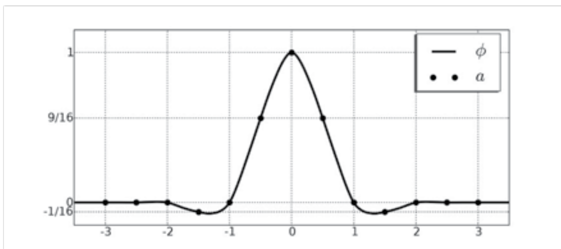
where p is any polynomial of degree $\leq 2n-1$. Also, the values of the refinable function at the half-integers are the values of the mask

$$\phi\left(\tfrac{j}{2}\right) = a_j, \quad j\in\mathbb{Z}. \qquad (18)$$

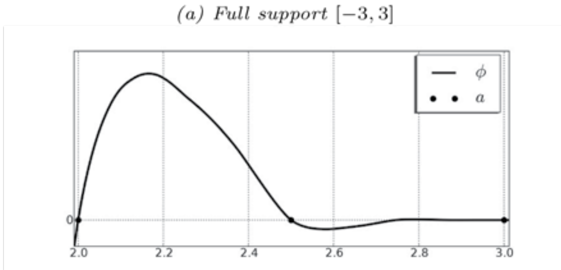
Finally, given the initial control points c, the limiting curve f of the Dubuc-Deslauriers subdivision scheme is given in terms of the refinable function as

$$f(x) = \sum_{j\in\mathbb{Z}} c_j \phi(x-j), \quad x\in\mathbb{R}. \qquad (19)$$

Some of these properties are illustrated in Figure 3 below.



(a) Full support $[-3,3]$



(b) Zoomed to $[2,3]$

Figure 3: Dubuc-Deslauriers refinable function ϕ and mask a.

The convergence of a subdivision scheme S_a ensures the existence of an associated refinable function since by choosing the original control points as the delta sequence $c=\delta=\{\delta_{0,j} : j\in\mathbb{Z}\}$ in Equation 4 the limit curve will be $f(x)=\sum_{j\in\mathbb{Z}}\delta_{0,j}\phi(x-j)=\phi(x),\ x\in\mathbb{R}$.

The converse of this statement is also true for interpolatory subdivision schemes. But for non-interpolatory subdivision schemes there are refinable functions for which the associated subdivision scheme is divergent, as shown in [6].

2.3. *A modified subdivision scheme for finite sequences*

The algorithms for bi-infinite sequences, as described in the previous sections, are applied mainly in the case of periodic sequences. For finite sequences these algorithms must be modified to accommodate the boundaries. Here we consider a method of adapting the Dubuc-Deslauriers subdivision scheme of Section 2.1 to the situation where the initial sequence c is finite.

The construction of the mask for finite sequences follows along similar lines as for the infinite case. The difficulty is that some of the values of the polynomial p in Equation 7 lie outside the finite domain and need to be supplied. This implies that an alternative mask needs to be constructed in the vicinity of the boundary. Following [7] and [8], we fit a polynomial of degree $2n-1$ to the $2n$ points next to, and including the boundary. Evaluating the resulting Lagrange polynomials at the half-integers next to the boundary yields the desired mask. The modified scheme for the left hand boundary ($j = 0, 1, \dots$) is given by

$$\begin{aligned} c_{2j}^{(r+1)} &= c_j^{(r)}, \\ c_{2j+1}^{(r+1)} &= \sum_{k\geq 0} a_{j,k} c_k^{(r)}, \end{aligned} \tag{20}$$

where for $j = 0, 1, \dots, n-2$ (close to the boundary),

$$a_{j,k} = \ell_{k-n+1}\left(j + \tfrac{1}{2} - n + 1\right) \tag{21}$$

for $k = 0, \dots, 2n-1$, and $a_{j,k} = 0$ for $k \notin \{0, 1, \dots, 2n-1\}$. For $j \geq (n-1)$ (away from the boundary)

$$a_{j,k} = \begin{cases} \ell_{k-j}(\frac{1}{2}), & k = -n+1+j, \dots, n+j, \\ 0, & \text{otherwise.} \end{cases}$$

If $n = 2$, for example Equation 21 gives

$$a_{0,k} = \ell_{k-1}(\tfrac{1}{2}) = \begin{cases} \frac{5}{16}, & k = 0, \\ \frac{15}{16}, & k = 1, \\ -\frac{5}{16}, & k = 2, \\ \frac{1}{16}, & k = 3, \\ 0, & \text{otherwise,} \end{cases}$$

and for $j \geq 1$ the mask is the same as before (see 12),

$$a_{j,k} = \begin{cases} -\frac{1}{16}, & k = 0, 3 \\ \frac{9}{16}, & k = 1, 2 \\ 0, & \text{otherwise.} \end{cases}$$

In the presence of a right hand boundary, the mask has to be modified in the same way as the left hand modifications, with the order reversed.

It is shown in [8] that a set of refinable functions exists for this modified mask (defined similarly to the definition Equation 13). The boundary modifications of the refinable function are illustrated in Figure 4. The existence of a set of refinable functions for the

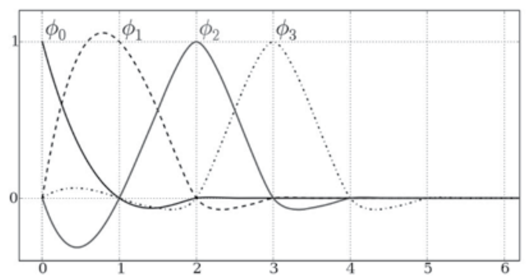


Figure 4: The refinable functions associated with the boundary-modified mask with $n = 2$

boundary-modified subdivision scheme allows one to construct wavelets for finite intervals. It is remarkable that these wavelets have finite decomposition and reconstruction sequences [8].

Next we discuss the so-called corner-cutting subdivision schemes.

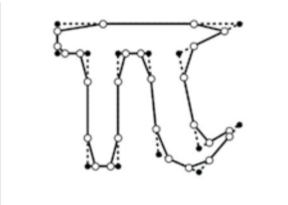
3. CORNER-CUTTING SUBDIVISION

In the case where the original control points contain noise we would not want to use an interpolatory subdivision scheme directly, but rather include some smoothing in the approximation. Since the limit curve of corner-cutting subdivision schemes does not pass through the control points, which amounts to some smoothing, corner-cutting subdivision schemes are better suited to this approximation problem. In this section we discuss one class of corner-cutting subdivision schemes, namely the de Rham-Chaikin scheme [9] and its generalization, the Lane-Riesenfeld scheme [10].

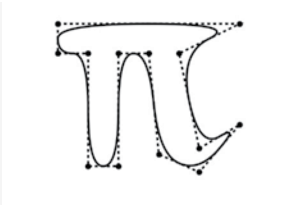
The only difference between the corner-cutting and the interpolatory schemes discussed in Section 2 lies in the choice of the mask of the operator in Equation 2. Conceptually, masks with positive entries generate corner-cutting subdivision schemes, since new control points are a weighted average of the old control points. The de Rham-Chaikin mask is given by

$$a_j = \frac{1}{4}\begin{pmatrix} 3 \\ j \end{pmatrix}, \quad j = 0, \dots, 3; \tag{22}$$

the corner-cutting property of this mask is illustrated in Figure 5.



(a) control points: original (•), after one iteration (○)



(b) original control points (•), control polygon after 6 iterations (–)

Figure 5: Subdivision with the de Rham-Chaikin mask Equation 22

Subdivision with the de Rham-Chaikin mask is convergent [9, 11], so that we are guaranteed the existence of a refinable function. This refinable function turns out to be the B-spline of degree 2 (order 3), denoted by B_2. Accordingly, the limit curve of this subdivision scheme is the quadratic spline

$$f(x) = \sum_j c_j^{(0)} B_2(x-j), \quad x \in \mathbb{R}. \tag{23}$$

The generalization of this scheme is known as the Lane-Riesenfeld scheme of order m. The Lane-Riesenfeld scheme of order m has mask

$$a_j^{(m)} = \frac{1}{2^{m-1}} \begin{pmatrix} m \\ j \end{pmatrix}, \quad j = 0, \ldots, m \tag{24}$$

and its limit curve is the spline of degree $m-1$ (see [10]) defined by

$$f(x) = \sum_j c_j^{(0)} B_{m-1}(x-j) \tag{25}$$

where B_{m-1} is the B-spline of degree $m-1$. Note that the smoothness of the limiting curve increases with m.

Note also that the Lane-Riesenfeld mask has finite support, i.e.

$$a_j^{(m)} = 0, \quad j \notin [0, m], \tag{26}$$

and that the mask elements within the support are all positive, i.e.

$$a_j^{(m)} > 0, \quad j \in [0, m]. \tag{27}$$

General results for finitely supported positive masks can be found in [12, 1].

All that remains to be done in the Lane-Riesenfeld example of corner-cutting subdivision is to modify the scheme in the presence of boundaries. The problem is the same as before—we need to supply missing values at the boundary. A very simple procedure is to repeat the boundary values as many times as needed. It turns out that this again leads to a a set of refinable functions, this time splines with multiple knots at the boundary. Thus the boundary-modified scheme again converges to a spline of the same degree as defined by the interior mask.

An example of these modified refinable functions is shown in Figure 6 and of the boundary-modified subdivision

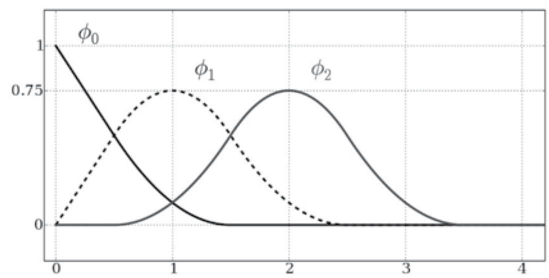
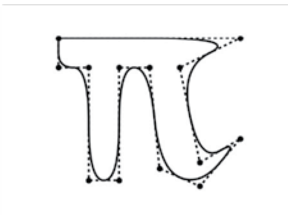
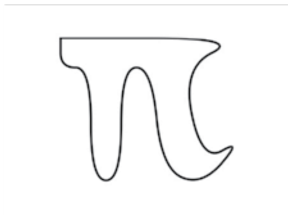


Figure 6: Boundary modified de Rham-Chaikin refinable functions

in Figure 7. Note we have chosen the top left hand control point as the first and last control points, hence we get a sharp corner at this point.



(a) original control points (•) and control polygon after 6 iterations (–)



(b) control polygon after 6 iterations

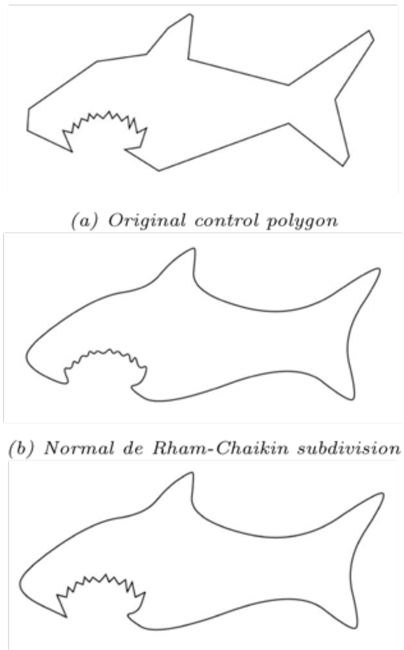
Figure 7: Boundary modified de Rham-Chaikin subdivision

Next consider the control polygon of a shark shown in Figure 8(a). Here we applied the standard de Rham-Chaikin, iterated to convergence, and obtained the sorry-looking shark of Figure 8(b)—a shark with blunt teeth is no shark at all. Doubling the control points defining the teeth results in the much happier-looking shark of Figure 8(c).

4. EXAMPLES

In this section we apply interpolatory and corner-cutting subdivision schemes to a few practical problems. The first example is a dynamic signature obtained via a digitising tablet. Figure 9(a) shows the original signature and the discretisation effects of this particular tablet should be obvious.

Since the samples are connected by straight lines, Fig-



(a) Original control polygon

(b) Normal de Rham-Chaikin subdivision

(c) de Rham-Chaikin subdivision with double points defining the teeth

Figure 8: Keeping corners by doubling control points

ure 9(a) can also be viewed as an example of a Dubuc-Deslauriers subdivision scheme with $n = 1$. Figure 9(b) shows the result of applying successive Dubuc-Deslauriers with $n = 2$ and downsampling. It is clearly smoother than Figure 9(a) but not as smooth as the de Rham-Chaikin corner-cutting subdivision and downsampled curve shown in Figure 9(c).

The next example increases the resolution of an image through subdivision. (Do not confuse this interpolation procedure with super resolution techniques where the resolution is increased by extracting additional information from multiple images.) Figure 10(a) shows the original image with a subsampled version shown in Figure 10(b).

We now apply subdivision to the subsampled version in an effort to recover the original. Figure 10(c) and (d) show the results of using the interpolatory Dubuc-Deslauriers scheme with $n = 2$ and the corner-cutting de Rham-Chaikin subdivision schemes, respectively. It is left to the reader to decide which one of the two schemes provide the more acceptable results.

5. SURFACE SUBDIVISION

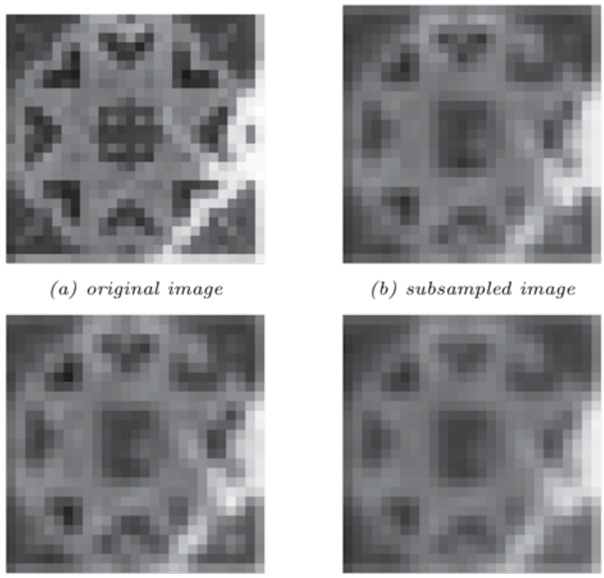
In the preceding sections we have limited our discussion to subdivision of curves. In this section we present the Doo-Sabin subdivision scheme [13] as an example of surface subdivision. Examples of other subdivision schemes are described in Catmull and Clark [14], and Loop [15]. For a good introduction see e.g. [16].

The Doo-Sabin scheme is based on uniform quadrilateral faces. It is a vertex split method that is based on biquadratic B-spline subdivision.

(a) original Signature

(b) interpolatory subdivision *(c) corner-cutting subdivision*

Figure 9: Smoothing a signature



(a) original image *(b) subsampled image*

(b) interpolatory subdivision *(c) corner-cutting subdivision*

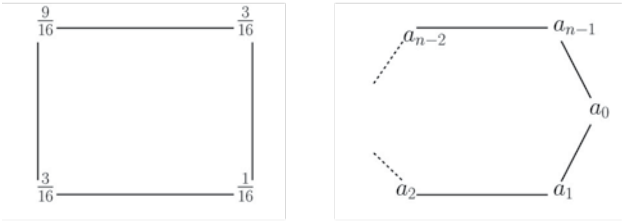
Figure 10: Smoothing an image (detail)

This scheme uses only one mask for all quadrilateral faces, shown in Figure 11(a). This mask is fitted to each face with the weights used cyclically to result in four children vertices per quadrilateral face.

Since some faces in the mesh will not be quadrilateral this mask will not always fit the faces. For the extraordinary faces (non-quadrilateral faces) we use a variable mask: for a face with n vertices we use the mask in Figure 11(b) with

$$a_i = \frac{1}{4n} \begin{cases} n+5, & i = 0, \\ 3 + 2\cos(\frac{2i\pi}{n}), & i \in \{1, \dots, n-1\}. \end{cases} \quad (28)$$

Notice that the mask for the extraordinary faces re-

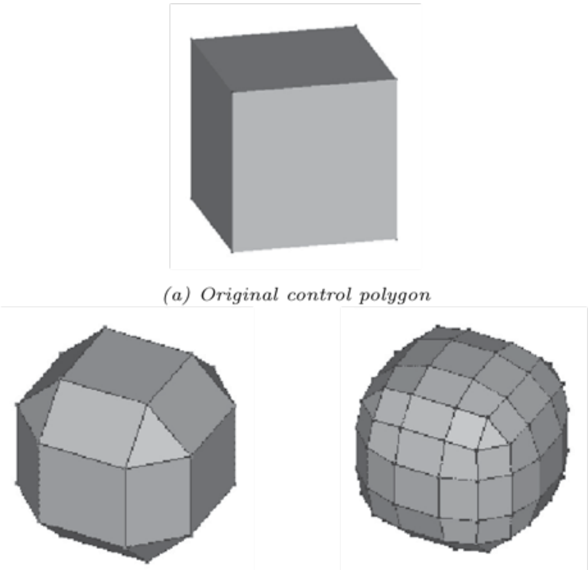


(a) mask for ordinary faces *(b) mask for extraordinary faces*

Figure 11: Doo-Sabin Subdivision scheme masks

duces to the mask for an ordinary face when $n = 4$.

In Figure 12, the first two iterations of this subdivision scheme are shown when applied to a unit cube.



(a) Original control polygon

(b) Mesh after first iteration *(c) Mesh after second iteration*

Figure 12: Doo-Sabin subdivision of a cube

Once again the nature of the subdivision scheme and the limiting surface depends entirely on the choice of mask. For surface subdivision using triangular meshes, see [15].

6. CONCLUSION

In this paper a brief overview of subdivision schemes for curves was given. The main ideas were explained for curves. In particular, interpolatory and corner-cutting schemes were discussed and the necessary boundary modifications for finite sequences were derived. The generalization to surfaces was briefly discussed. The numerical examples demonstrate the power of these methods to generate smooth curves and surfaces from a limited number of control points.

7. REFERENCES

[1] C. A. Micchelli, *Mathematical Aspects of Geometric Modeling.* CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA, 1995.

[2] S. Dubuc, "Interpolation through an iterative scheme," *J. Math. Anal. Appl.*, vol. 114, pp. 185–204, 1986.

[3] G. Deslauriers and S. Dubuc, "Symmetric iterative interpolation processes," *Constr. Approx.*, vol. 5, no. 1, pp. 49–68, 1989.

[4] D. Knuth, *Digital Typography.* CSLI Publications, Stanford, California, 1999.

[5] C. A. Micchelli, "Interpolatory subdivision schemes and wavelets," *J. Approx. Theory*, vol. 86, pp. 41–71, 1996.

[6] M. Neamtu, "Convergence of subdivision versus solvability of refinement equations," *East Journal of Approximations*, vol. 5, no. 2, pp. 183–210, 1999.

[7] D. L. Donoho, "Smooth wavelet decompositions with blocky coefficient kernels," in *Recent Advances in Wavelet Analysis* (L. L. Schumaker and G. Webb, eds.), vol. 3 of *Wavelet Analysis and Its Applications*, pp. 259–308, Academic Press, Boston, 1994.

[8] J. M. de Villiers, K. M. Goosen, and B. M. Herbst, "Dubuc–Deslauriers subdivision for finite sequences and interpolation wavelets on an interval," *SIAM J. Math. Anal.*, vol. 35, no. 2, pp. 423–452, 2003.

[9] G. de Rham, "Sur une courbe plane," *Journal de Mathématiques Pures et Appliquées*, vol. 35, pp. 25–42, 1956.

[10] J. M. Lane and R. F. Riesenfeld, "A theoretical development for the computer generation and display of piecewise polynomial surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 35–46, 1980.

[11] G. M. Chaikin, "An algorithm for high-speed curve generation," *Computer Graphics and Image Processing*, vol. 3, pp. 346–349, 1974.

[12] C. A. Micchelli and H. Prautzsch, "Refinement and subdivision for spaces of integer translates of a compactly supported function," in *Numerical Analysis* (D. F. Griffiths and G. A. Watson, eds.), pp. 192–222, Longman, London, 1987.

[13] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, pp. 356–360, September 1978.

[14] E. Catmull and J. Clark, "Recursively generated b-spline surfaces on arbitrary topological meshes," *Computer-Aided Design*, vol. 10, pp. 350–355, September 1978.

[15] C. Loop, "Smooth subdivision surfaces based on triangles," Master's thesis, University of Utah, Department of Mathematics, 1987.

[16] J. O. M. Larsen and C. K. Madsen, "Subdivision surfaces," tech. rep., Department of Mathematics, Technical University of Denmark, July 2001.