Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Smooth Subdivision of Tetrahedral Meshes

S. Schaefer[1] J. Hakenberg[1] J. Warren[1]

[1] Rice University

Abstract

We describe a new subdivision scheme for unstructured tetrahedral meshes. Previous tetrahderal schemes based on generalizations of box splines have encoded arbitrary directional preferences in their associated subdivision rules that were not reflected in tetrahderal base mesh. Our method avoids this choice of preferred directions resulting a scheme that is simple to implement via repeated smoothing. In an extended appendix, we analyze this tetrahedral scheme and prove that the scheme generates C^2 deformations everywhere except along edges of the tetrahedral base mesh. Along edges shared by four or more tetrahedra in the base mesh, we present strong evidence that the scheme generates C^1 deformations.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Given a base mesh p^0, subdivision is a recursive process that defines increasingly refined meshes via a relation of the form

$$p^{k+1} = Sp^k.$$

If the subdivision process S is chosen correctly, the limit mesh p^∞ is guaranteed to be a smooth mesh that approximates the base mesh p^0. Using subdivision has become popular for geometric modeling because the subdivision process places no restriction on the topological connectivity of the base mesh.

While most work on subdivision has focused on surface meshes, we consider the problem of subdividing volumetric meshes. Perhaps the most obvious questions to ask concerning volumetric subdivision is why bother with building such schemes. Typically, volumetric subdivision schemes have been proposed as a means to define deformations. However, the existence of simple schemes for tensor product volumetric meshes such as free-form deformations [SP86] reduces this question to why subdivision scheme for unstructured meshes are important.

Figure 1 shows an application of subdivision to the problem of image deformation that illustrates the superiority of unstructured methods. In this case, the image being deformed is a cross-section of a mouse brain where the pixel intensities represent the cell density in different anatomical (colored) regions of the brain. On the left, the image has been covered by a uniform base mesh. Subdividing this quad mesh using bi-cubic subdivision yields a C^2 mesh that defines a smooth parameterization of the image. Perturbing the vertices of the base mesh in the region of the cerebellum (dark folds) induces a corresponding deformation of the underlying image. On the right, the image has been covered by an unstructured quadrilateral mesh. A subset of the edges in this mesh have been creased to define a network of crease curves that partition the base mesh into anatomical regions. Now, this quadrilateral base mesh is subdivided using Catmull-Clark subdivision to define a smooth parameterization of the underlying image. Perturbing the vertices of the quadrilateral base mesh induces deformations that are restricted to a single anatomical region. Thus, the use of unstructured mesh allows the construction of deformations with much finer control than those built using tensor product methods.

1.1. Previous work

While previous work on subdivision of unstructured volumetric meshes has been limited, there are a few papers that have addressed this problem. MacCracken and Joy [MJ96] developed one of the first volumetric subdivision schemes. This scheme was developed primarily to define deformations based on unstructured hexahedral meshes. Unfortunately, the subdivision rules proposed in the paper were de-

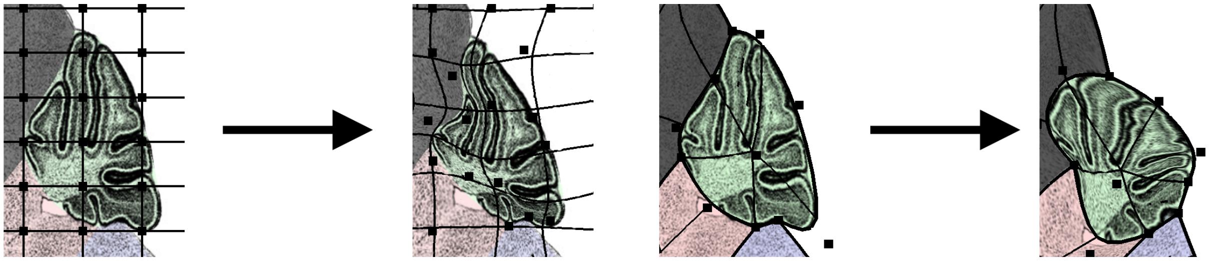

Figure 1: *Close-up of the cerebellum on a cross-section of a mouse brain. Initial embedding in a uniform grid and its deformation using free-form deformations (left). Subdivision surface that models the boundaries of the different anatomical regions and the explicit deformation of those boundaries (right).*

veloped in a somewhat ad-hoc manner making any type of proof of smoothness for the scheme very difficult. Later, Bajaj et al [BSWX02] developed different subdivision rules for hexahedral meshes, which generated deformations that were provably smooth everywhere except at vertices of the hexahedral base mesh.

Both of these schemes used hexahedra (topological cubes) as their volumetric elements. Unfortunately, building unstructured meshes of hexahedra that conform to specific boundary shapes can be difficult. Traditionally, mesh generation methods generate unstructured meshes of tetrahedra instead. The most relevant piece of previous work is a subdivision scheme for unstructured tetrahedral meshes proposed by Chang et al [CMQ02]. In that paper, the authors build subdivision rules for unstructured tetrahedral meshes by generalizing the subdivision rules for a particular class of trivariate box-splines.

While this approach was successfully used by Loop [Loo87] to generalize the subdivision rules for the C^2 three-direction quartic box splines to unstructured triangular meshes, using trivariate box splines to generate subdivision rules for unstructured tetrahedral grids is much more difficult. The drawback of the subdivision rules proposed in Chang et al's is that these rules encode a preferred direction in each tetrahedron of the base mesh. (Section 2.1 will elaborate on this point.) This directional preference makes implementing the Chang et al scheme tricky and proving any results concerning the smoothness of the scheme extremely difficult.

Contributions

In contrast to Chang et al, we develop volumetric subdivision rules for unstructured tetrahedral meshes that avoid the assumption of any preferred direction in the base mesh. This construction also generalizes the bivariate case and leads to a trivariate scheme with two important properties:

- The scheme is simple to implement in terms of linear subdivision and smoothing.
- The deformations induced by the scheme are provably C^2 everywhere except along edges of the base mesh. Along edges shared by four or more tetrahedra, we present strong evidence that the resulting deformations are C^1.

The body of the paper presents the tetrahedral scheme and considers several of its applications with no accompanying theoretical analysis. In an extended appendix, we perform a mathematical analysis of the smoothness of the scheme using a combination of regularity analysis (Reif [Rei95]) and spectral analysis (Levin/Levin [LL03]).

2. A tetrahedral subdivision scheme

Our proposed scheme is a combination of linear subdivision followed by a smoothing pass. This structure is similar to that of the several schemes proposed for subdividing surface meshes [BSWX02, Sta01, ZS01]. As in the bivariate case, implementing our scheme is quite simple and does not require neighbor finding or mesh traversal algorithms. To illustrate the ease of implementation, we provide pseudocode for the smoothing pass at the end of this section.

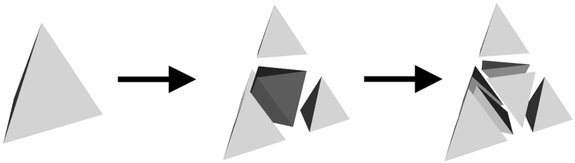

Figure 2: *Splitting a tetrahedra generates an octahedron in the middle. Splitting the octahedron into tetrahedra requires the choice of a diagonal.*

2.1. Linear subdivision

To perform linear subdivision on a mesh of tetrahedra, we define a split on a single tetrahedron, which is then applied to all tetrahedra in the mesh. Given a tetrahedron, we insert new vertices at the midpoints of each edge and connect the vertices together to form four new tetrahedra at the corners of the original tetrahedron. Chopping these four children off the corners of the parent tetrahedron leaves an octahedron

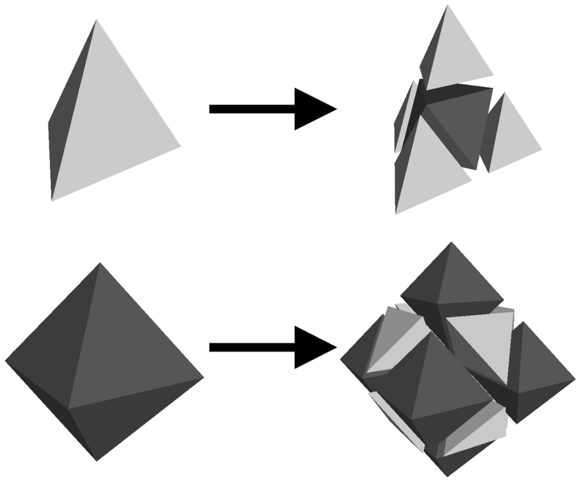

Figure 3: *Linear subdivision splits a tetrahedron into four tetrahedra and an octahedron (top). An octahedron is split into six octahedra and eight tetrahedra (bottom).*

(see figure 3 top). (Note that performing this corner chopping on a triangle yields a triangle making linear subdivision for triangular meshes much easier.)

At this point, we are faced with a dilemma. We can either split the octahedron into four tetrahedra by choosing a diagonal for the octahedron (see figure 2) or leave the octahedron alone and develop an analog of linear subdivision for octahedra. At first glance, splitting the octahedron along a diagonal might seem like a simplifying choice. In reality, this choice leads to substantial complications during any attempt to analyze the smoothness of the associated subdivision scheme. This choice of diagonal cause the resulting tetrahedral mesh to contain a preferred direction associated with the choice of diagonal. To generate a provably smooth subdivision scheme, this diagonal must be inherited during linear subdivision. More crucially, each tetrahedron in the base mesh must be assigned such a diagonal. Any type of smoothness analysis that considers the interface between two tetrahedra in the base mesh must enumerate all possible choices for this diagonal.

Given that our goal is to create a scheme that contains no preferred direction and is simple enough to prove smoothness results about, we do not choose a diagonal for the middle octahedron and split a tetrahedron into four new tetrahedra and an octahedron (see figure 3). Since we have introduced an octahedron into the volumetric mesh, our subdivision scheme is not simply a tetrahedral subdivision scheme, but a tetrahedral/octahedral subdivision scheme. Therefore, we must define a refinement rule for octahedra as well.

To refine an octahedron, we insert vertices at the midpoints of each edge on the octahedron and at the centroid of the octahedron, which is formed by averaging all of the vertices of the octahedron together. Next, we connect the vertices together to form six new octahedra (corresponding to the six vertices of the original octahedron) and eight new tetrahedra (corresponding to the eight faces of the original octahedron). The entire refinement process is illustrated in figure 3.

While Chang et al's tetrahedral scheme is similar to ours in that it does not topologically split the octahedron, their scheme generates subdivision rules that encode a preferred diagonal along the octahedron. This preferred diagonal is a natural result of their use of trivariate box splines in generating their subdivision rules. Due to the existence of a preferred diagonal, Chang et al's scheme is guaranteed to be smooth only on the interior of each tetrahedron in the base mesh. In particular, Chang et al make no attempt to analyze the smoothness of their scheme across the face shared by two tetrahedra in the base mesh.

The need for such face/face analysis is somewhat surprising and was not even recognized by Chang et al. This failure is understandable since a subdivided triangular mesh is uniform along the interior of edges in the base mesh. Similarly, a subdivided hexahedral mesh is uniform along the interior of quad faces of the base mesh. Unfortunately, a subdivided, unstructured tetrahedral mesh is *not* uniform across the interior of triangular faces of the base mesh. Thus, substantial care must be used in designing the subdivision rules of the scheme if one hopes to construct a scheme that is provably smooth across these faces. In the appendix, we use the joint spectral radius techniques of Levin/Levin to prove that our scheme is C^2 across the interior of these faces.

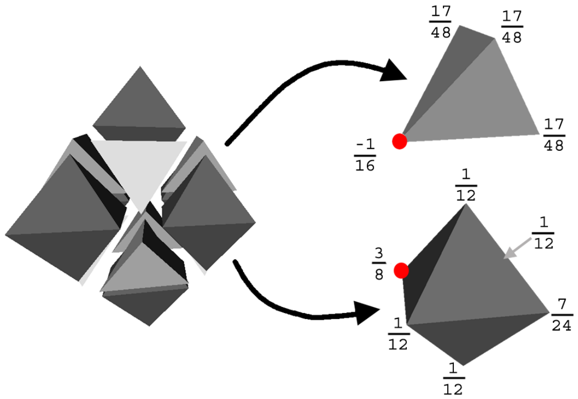

Figure 4: *Centroid masks for tetrahedra/octahedra. The highlighted vertex is the vertex being repositioned by smoothing.*

2.2. Smoothing

After linear subdivision, we perform a smoothing pass over the tetrahedral/octahedral mesh to reposition each vertex.

```
// input: T is list of cells, p is an array of vertex positions
// a cell is a list of indices into p
$newP \leftarrow 0$
$val \leftarrow 0$
for each $T_i$
    if ( $T_i$ is a tetrahedron )
        $newP[T_i] += \begin{pmatrix} \frac{-1}{16} & \frac{17}{48} & \frac{17}{48} & \frac{17}{48} \\ \frac{17}{48} & \frac{-1}{16} & \frac{17}{48} & \frac{17}{48} \\ \frac{17}{48} & \frac{17}{48} & \frac{-1}{16} & \frac{17}{48} \\ \frac{17}{48} & \frac{17}{48} & \frac{17}{48} & \frac{-1}{16} \end{pmatrix} .p[T_i]$
    else // must be an octahedron
        $newP[T_i] += \begin{pmatrix} \frac{3}{8} & \frac{1}{12} & \frac{1}{12} & \frac{7}{24} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{3}{8} & \frac{1}{12} & \frac{1}{12} & \frac{7}{24} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{3}{8} & \frac{1}{12} & \frac{1}{12} & \frac{7}{24} \\ \frac{7}{24} & \frac{1}{12} & \frac{1}{12} & \frac{3}{8} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{7}{24} & \frac{1}{12} & \frac{1}{12} & \frac{3}{8} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{12} & \frac{7}{24} & \frac{1}{12} & \frac{1}{12} & \frac{3}{8} \end{pmatrix} .p[T_i]$
    $val[T_i]$++
for each $newP_i$
    $newP_i$ /= $val[i]$
return mesh of $\{T, newP\}$
```

Figure 5: *Smoothing pass for tet/oct subdivision*

For each vertex in the mesh after linear subdivision, we find each volumetric cell (tetrahedron or octahedron) containing that vertex. Then we compute the weighted centroids shown in figure 4 for each cell. For tetrahedra, this centroid computes $\frac{-1}{16}$ of the vertex being repositioned and $\frac{17}{48}$ of the edge adjacent vertices. To generate the centroid for octahedra we take $\frac{3}{8}$ of the vertex being repositioned, $\frac{1}{12}$ of the edge adjacent vertices and $\frac{7}{24}$ of the cell-adjacent vertex. We then average all of these centroids together to obtain the new location of the repositioned vertex. Despite the fact that there is a negative weight in the centroid mask for tetrahedra, the subdivision rules produced by combining linear subdivision and smoothing use only convex combinations.

Since this smoothing pass is described only in terms of centroid masks, it yields a very simple implementation. Given an unstructured tetrahedra/octahedra mesh, we first apply linear subdivision. This operation can be implemented on a cell by cell basis. For smoothing, we initialize the vertices of a mesh with the same topology as the input to be identically 0. Then, for each cell in the mesh, we compute the centroid mask of figure 4 in all possible orientations and add that quantity to the vertex to be repositioned for that orientation. Finally, we divide each vertex by its valence (the number of cells containing that vertex). Figure 5 illustrates pseudocode for the smoothing pass. This description requires no neighbor finding in the mesh or external data structures to traverse the mesh and is quite easy to implement.

We can also incorporate sharp features where we alter the continuity of the volume to be C^0 easily using the method described by Hoppe et al [HDD*94]. Figure 8 shows a cylindrical volume with a crease surface defined by Loop subdivision and crease edges, which form B-splines, around the top and bottom of the cylinder.

3. Application to deformations

As alluded to in the introduction, volumetric subdivision schemes find their main use in generating smooth volumetric deformations. Given a volume R, a volumetric deformation f maps points x in R to new points $f(x)$ in $f(R)$. The deformation f is C^k continuous if each coordinate function comprising f can be expressed locally as the graph of a function with k continuous derivatives.

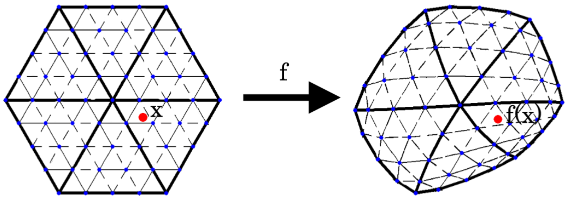

Figure 6: *Subdivision defines a map f between points in a rest configuration to a deformed configuration.*

To construct volumetric deformations, we use the basic technique described in MacCracken and Joy [MJ96]. Given a base mesh p^0, we define R to be the volume spanned by the limit mesh p^∞. If the mesh p^∞ forms a one-to-one covering of R, we can express each point x in R as a unique point on the limit mesh p^∞ of the form $\sum_i \alpha_i p_i^0$ where the p_i^0 are vertices of the base mesh p^0 (see figure 6). Perturbing the vertices of the base mesh to form a new mesh $\hat{p}^0$ defines an associated deformation f of the form

$$f(x) = \sum_i \alpha_i \hat{p}_i^0. \qquad (1)$$

In the attached appendix, we show that the deformations induced by our tetrahedral subdivision scheme are provably C^2 everywhere except along edges of the base mesh. Along edges shared by four or more tetrahedra, we hypothesize the scheme is C^1 and provide strong evidence to support this hypothesis.

Figure 7 shows an application of our method to the problem of deforming a dinosaur skeleton. First, we embed the skeleton in a tetrahedral base mesh p^0. Next, we perform several rounds of subdivision on the base mesh to form a refined mesh p^k. Each new vertex inserted in the tetrahedral/octahedral mesh p^k is represented as an convex combination of the vertices of the base mesh p^0. Then, for each vertex x of the skeleton, we find the tetrahedra or octahedra in the refined mesh p^k that contains that vertex and compute

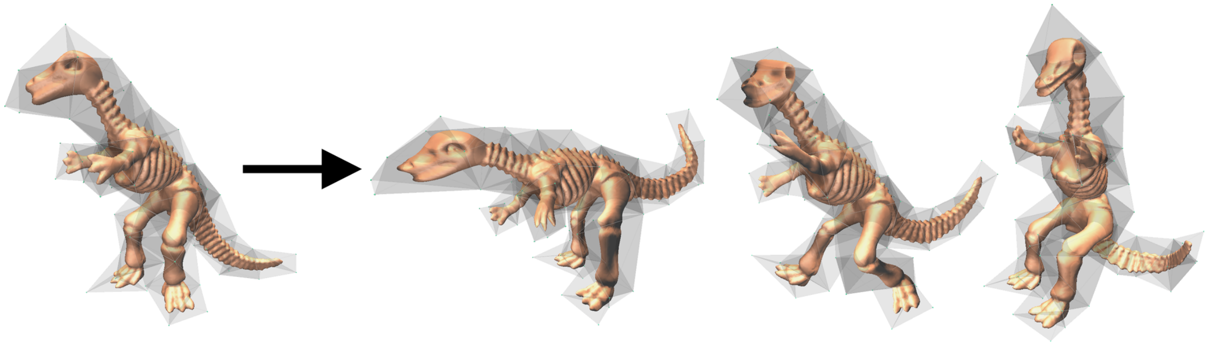

Figure 7: *Surface deformation via tetrahedral subdivision. Initial shape shown (left) and three deformed poses (right).*

the barycentric coordinates of that vertex with respect to its enclosing cell. Since all vertices of the cell are convex combinations of vertices of p^0, the skeleton vertex x can then be represented as a convex combination of the vertices of p^0. Perturbing the vertices of the base mesh forms a new base mesh $\hat{p}^0$ that defines a deformation $f(x)$ of the vertices of the skeleton as given in equation 1.

Figure 7 illustrates several different deformations of the dinosaur model. Since we use an unstructured grid of tetrahedra, we can encase the surfaces to be deformed in far fewer volumetric elements than would be required by free-form deformations using structured grids. Therefore, this sparse embedding yields deformations that require relatively few tetrahedral vertices and can be performed in real-time.

Figure 8 depicts another example in which a 3D test pattern is deformed using our scheme. The base mesh p^0 is a tetrahedral mesh approximating the shape of a cylinder. To generate the sharp circular edges along the top and bottom of the cylinder, we have creased the appropriate edges of the cylinder. The left part of the figure shows the tetrahedral mesh in wireframe. The middle and right portions of the figure show the 3D test pattern before and after perturbation of the base mesh.

4. Conclusions

We have presented a simple subdivision scheme for unstructured tetrahedral mesh. This scheme consists of linear subdivision followed a smoothing pass. Implementing the pass of the scheme requires only a standard topological mesh representation without the need for any auxiliary adjacency information. The key to the simplicity of the scheme is the symmetric treatment of the octahedron generated by chopping off the corners of a tetrahedron. As we show in the appendix, this choice makes smoothness analysis possible.

5. Acknowledgements

This work was supported by NSF grant ITR-0205671.

References

[BSWX02] Bajaj C., Schaefer S., Warren J., Xu G.: A smooth subdivision scheme for hexahedral meshes. In *The Visual Computer* (2002), vol. 18, pp. 409–420. 2

[CMQ02] Chang Y.-S., McDonnell K. T., Qin H.: A new solid subdivision scheme based on box splines. In *Proceedings of the seventh ACM symposium on Solid modeling and applications* (2002), ACM Press, pp. 226–233. 2

[HDD*94] Hoppe H., DeRose T., Duchamp T., Halstead M., Jin H., McDonald J., Schweitzer J., Stuetzle W.: Piecewise smooth surface reconstruction. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994), ACM Press, pp. 295–302. 4

[LL03] Levin A., Levin D.: Analysis of quasi uniform subdivision. *Applied and Computational Harmonic Analysis 15(1)* (2003), 18–32. 2, 7, 8

[Loo87] Loop C.: Smooth subdivision surfaces based on triangles, 1987. 2

[MJ96] MacCracken R., Joy K. I.: Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM Press, pp. 181–188. 1, 4

[Rei95] Reif U.: A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design 12*, 2 (1995), 153–174. 2, 8

[SP86] Sederberg T. W., Parry S. R.: Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (1986), ACM Press, pp. 151–160. 1

[Sta01] Stam J.: On subdivision schemes generalizing uniform b-spline surfaces of arbitrary degree. In *Computer Aided Geometric Design* (2001), vol. 18, pp. 383–396. 2

[War95] Warren J.: Subdivision methods for geometric design. unpublished manuscript, 1995. 7

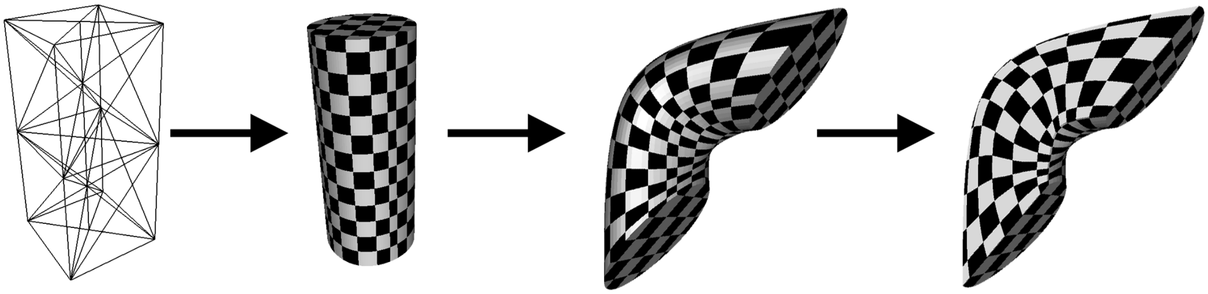

Figure 8: *Initial set of tetrahedra, subdivided surface, deformed surface and cut interior. Parameter lines are smooth after deformation.*

[ZS01] ZORIN D., SCHRODER P.: A unified framework for primal/dual subdivision schemes. In *Computer Aided Geometric Design* (2001), vol. 18, pp. 429–454. 2

Appendix A: Smoothness analysis

Given a tetrahedral base mesh p^0, we consider the smoothness of a deformation $f(x)$ induced by a perturbation of the base mesh. In particular, our smoothness analysis considers four cases:

- x lies in the interior of a tetrahedron of the base mesh,
- x lies on the interior of a face shared by two tetrahedra of the base mesh,
- x lies on the interior of an edges shared by several tetrahedra of the base mesh,
- x lies at a vertex of the base mesh.

Interior of a base tetrahedron

To begin our analysis, we first consider the structure of the uniform mesh generated by linearly subdividing a single tetrahedron repeatedly. If this base tetrahedron has vertices of the form $(1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1)$, k rounds of linear subdivision generate a uniform 3D mesh whose vertices have barycentric coordinates of the form $\frac{1}{2^k}(i_0,i_1,i_2,i_3)$ where the i_j are non-negative integers that sum to 2^k. (The embedding of the base tetrahedron in the plane $x_0+x_1+x_2+x_3=1$ allows the coordinates to be treated symmetrically and avoids the use of any preferred direction in our construction.)

Relaxing the restriction that the coordinates of the mesh vertices are non-negative yields a sequence of infinite uniform meshes M^k associated with the subdivision process. Unfortunately, the scaling relation between consecutive meshes M^k and M^{k+1} is subtle due to the use of barycentric coordinates. However, if we translate the base mesh M^0 to interpolate the origin, the resulting mesh M_c lies on the plane $x_0+x_1+x_2+x_3=0$ and has vertices of the form (i_0,i_1,i_2,i_3) where the i_j are integers whose sum is 0. After translation, linearly subdividing the mesh M_c yields a dilated mesh of the form $\frac{1}{2}M_c$.

For this scale-invariant mesh, we now consider the hat function $n(x)$ centered at the origin that is generated by linear subdivision with no smoothing. Based on the splitting rules for linear subdivision, $n(x)$ satisfies a scaling relation of the form

$$n(x)=n(2x)+\frac{1}{2}\sum_{j=1}^{12}n(2x-\delta_j)+\frac{1}{6}\sum_{j=1}^{6}n(2x-\gamma_j) \qquad (2)$$

where the vectors δ_j and γ_j define integer offsets in the uniform mesh M. These offsets correspond to vertices of M_c that lie in the one-ring of the origin (see the left portion of figure 9) and correspond to

$$\begin{aligned}\delta &= \text{Permutations}(1,-1,0,0)\\ \gamma &= \text{Permutations}(1,1,-1,-1)\end{aligned}$$

where Permutations yields a set without duplication (δ contains 12 elements while γ has 6). The right portion of figure 9 shows the 3D subdivision mask formed by the coefficients of equation 2.

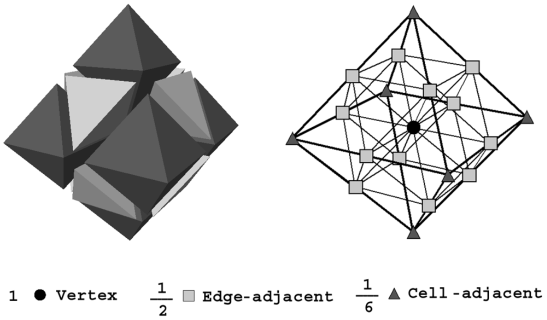

Figure 9: *The subdivision mask for the linear subdivision on the uniform grid M.*

We next consider the effect of the smoothing pass on the mesh formed by linear subdivision. If we apply the

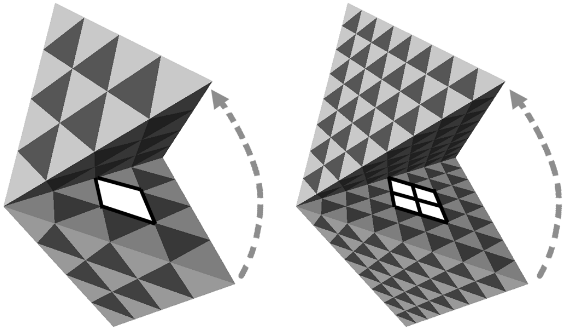

Figure 10: *Two face-adjacent tetrahedra subdivided and opened at the shared face. Subdivision generates tet/tet and oct/oct pairs along that face.*

weighted-centroid averaging rules for this pass to the one-ring of the origin, the smoothing mask that results is also supported over the one-ring of origin and is exactly $\frac{1}{8}$ of the subdivision mask for linear subdivision. (The agreement is no coincidence as we chose the weights used in the smoothing pass to ensure this agreement.)

Now, the subdivision mask for the composite scheme formed by linear subdivision and smoothing is simply $\frac{1}{8}$ of the discrete convolution of the mask of figure 9 with itself. As shown in [War95], the discrete convolution of two subdivision masks yields a new subdivision mask whose associated basis function is the continuous convolution of the basis functions associated with each original mask. In our case, the convolution of the hat function $n(x)$ with itself is the basis function for our composite scheme. Since the continuous convolution of two C^0 functions ($n(x)$ with itself) is always a C^2 function, our scheme generates C^2 deformations on uniform meshes.

Faces of the base mesh

While the smoothness of our subdivision scheme on the interior of a base tetrahedron follows by appealing to convolution, we must verify smoothness of our scheme in the other cases using spectral methods suitable for analysis of non-uniform schemes. This difference is due to the fact that the mesh formed by applying linear subdivision to an unstructured tetrahedral mesh is uniform only on the interior of base tetrahedra and is no longer uniform across the faces of the base mesh. In particular, the uniform mesh M_c generated on the interior of a base tetrahedron has the property that every face in the grid is shared by a tet/oct pair. On the other hand, the infinite mesh M_f generated by applying linear subdivision to two face-adjacent tetrahedra consists of two copies of M_c joined along a triangular interface formed by tet/tet pairs and oct/oct pairs. (see figure 10.)

To prove that our scheme is C^2 on the mesh M_f, we use the joint spectral radius test originally developed by Levin/Levin [LL03] to analyze the smoothness of triangle/quad subdivision along the interface between triangles and quads. Note that the mesh structure in their triangle/quad analysis is similar to the mesh structure of M_f: two uniform meshes separated by planar interface.

If S is the subdivision matrix associated with M_f, we first compute the eigenvalues λ_j and eigenvectors z_j of the form $Sz_j = \lambda_j z_j$. The Levin/Levin C^2 smoothness test involves checking three conditions:

- First, check whether the eigenvalues λ_j (ordered in descending value) have the form
$$1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} > \dots \tag{3}$$
Note that the subdominant eigenvectors (z_1, z_2, z_3) reproduce the grid M_f. As a result, the eigenfunctions associated with these eigenvectors define a characteristic map that produces a one-to-one covering of space.
- Next, we check whether the eigenfunctions associated with eigenvectors $z_4, \dots, z_9$ are quadratic functions when plotted over the characteristic map.
- Finally, the joint spectral radius of the subdivision scheme must be less than $\frac{1}{4}$.

The first test is simple to check and involves only extraction of the eigenvalues and eigenvectors of the subdivision matrix. The second condition can be checked using quasi-interpolants as in Levin/Levin [LL03]. However, the joint spectral radius test requires more work.

Figure 10 (left) shows a portion of M_f with the interface between a tet/tet and oct/oct pair highlighted. This pair forms a patch on the face between two tetrahedra. To perform the joint spectral radius test, we must construct 4 subdivision matrices S_i that map the support of the patch on figure 10 (left) to the support of each of the four sub-quads formed after one round of subdivision (figure 10 right). Each sub-quad is a scaled and translated version of the original quad and yields a square subdivision matrix S_i.

After building the S_i, we then construct a diagonalizing matrix W using S_1 such that

$$\begin{aligned} W^{-1}S_1W &= \begin{pmatrix} \Lambda & C_1 \\ 0 & Y_1 \end{pmatrix} \\ W^{-1}S_iW &= \begin{pmatrix} \theta_i & C_i \\ 0 & Y_i \end{pmatrix} i \neq 1 \end{aligned}$$

where Λ is a diagonal matrix whose entries are the specified eigenvalues in equation 3 and θ_i is an upper triangular matrix that shares the same diagonal entries as Λ. W can be constructed using the eigenvectors in S_1 corresponding to the eigenvalues in Λ and the null space of those vectors.

Finally, to perform the joint spectral radius test, we compute

$$\rho^{[k]}(Y_1, ..., Y_4) = (\mathrm{Max} \| Y_{\varepsilon_k} Y_{\varepsilon_{k-1}} \dots Y_{\varepsilon_1} \|_\infty)^{\frac{1}{k}}$$

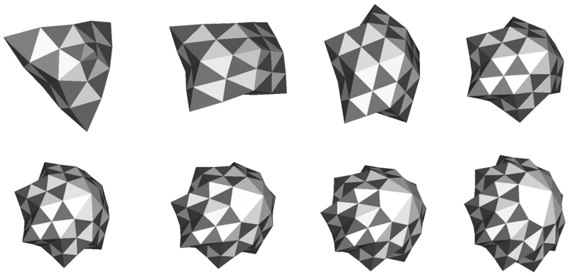

Figure 11: *Characteristic map for edges of valence* 3 *through* 10.

where $\epsilon_i \in \{1, \ldots, 4\}$. If $\rho^{[k]} < \frac{1}{4}$ for some k, then the scheme is C^2 over that extraordinary complex. For our four subdivision matrices we computed $\rho^{[9]} = 0.238$ and conclude that our scheme is C^2 along the face shared by two tetrahedra.

Edges of the base mesh

To analyze the smoothness of our scheme along edges of the base mesh is more difficult than the face case since the structure of the infinite mesh M_e form by subdividing n tetrahedra sharing a common edge depends on n. In practice, we know of no analysis technique capable of establishing the smoothness of our scheme along this edge. However, we hypothesize that a combination of the analysis methods of Levin/Levin [LL03] and Reif [Rei95] can be used to analyze the smooth of volumetric scheme in configurations of this type.

Given the subdivision matrix S for the mesh M_e, we hypothesize that the scheme is C^1 if:

- Its eigenvalues are of the form $1 > \lambda_1 \geq \lambda_2 \geq \lambda_3 > \ldots$
- The characteristic map formed by the eigenvectors z_1, z_2, z_3 is regular and injective.
- Finally, the joint-spectral radius of the scheme ($\rho^{[k]}(Y_1, Y_2)$) must be less than λ_3.

The core of our hypothesis is that the second condition allows the smoothness analysis to be reduced to the functional case used in the joint spectral radius test.

We have checked these three criteria for our scheme. Since the mesh M_e is parameterized by the number of tetrahedra sharing the edge of the base mesh, the subdivision matrices S will contain a block circulant structure that makes extraction of the eigenvectors z_1, z_2, z_3 as a symbolic function of n possible. However, it is unclear how the block circulant structure interacts with the joint spectral radius test. Therefore, we have numerically computed the characteristic map for edge valences 3 through 10 and visually inspected their shape (see figure 11).

To apply the joint spectral radius test, we construct subdivision matrices S_1, S_2 corresponding to one of two shifts along the edge. For our scheme, these matrices have eigenvalues of the form $1, \lambda_1, \lambda_1, \frac{1}{2}, \ldots$ and satisfy the joint spectral radius test for valences $10 \geq n > 3$. For $n = 3$, we could not find a k that satisfied the joint spectral radius condition. While this failure does not mean the scheme is C^0, we visually inspected the smoothness of the volumes using the 3D test patterns shown in figure 8 and the deformations produced are not smooth. For other valences $10 \geq n > 3$, the 3D test patterns undergoes visually smooth deformation.

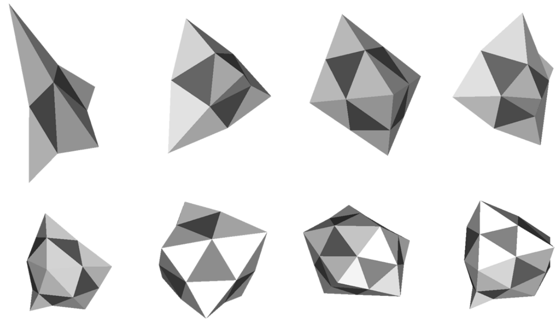

Figure 12: *Characteristic maps for several low valence configurations of tetrahedra around a vertex.*

Vertices of the base mesh

As for the edge case, we know of no analysis method for proving that our scheme is smoothness at a vertex of the base mesh. However, we again hypothesis that the conditions for the edge case suffice to establish smoothness at a vertex. In the vertex case, the third test is redundant since there is only a single matrix $S_1 = S$ used in computing the joint spectral radius.

Similar to the edge case, we only consider arbitrary packing of tetrahedra around a base vertex. Unfortunately, the simple parameterization by valence used in analyzing surface subdivision schemes is unavailable for volume schemes and no block circulant structures can be exploited to analyze the smoothness of the scheme. Therefore, we enumerated through all configurations of tetrahedra around a vertex for valences 4 through 10 and tested random configurations of tetrahedra for higher valences. Each configuration of tetrahedra passed the eigenvalue and characteristic map test. Figure 12 shows several characteristic maps produced for different configurations and number of tetrahedra around a vertex.