

Normal Control using N-adic Subdivision Schemes

G. Salomon[†], A. Leclercq[†], S. Akkouche[‡] and E. Galin[†]

[†]L.I.G.I.M

Université Claude Bernard Lyon 1
69622 Villeurbanne Cedex
France

[gsalomon|aleclercq|egalin]@ligim.univ-lyon1.fr

[‡]L.I.G.I.M

Ecole Centrale de Lyon
B.P. 163, 69131 Ecully Cedex
France

samir@ec-lyon.fr

Abstract

This paper presents simple and efficient deformation methods based on a new class of interpolating N-adic subdivision algorithms. Our N-adic scheme is a natural extension of a standard dyadic scheme – each face of the mesh is more generally divided into N^2 sub-faces – and geometric properties are similar.

This framework enables us to locally deform the surface using different tools by either modifying the direction of normals at the vertices of the control mesh, or twisting them.

Experiments show that the N-adic decomposition provides a more accurate control over deformations, and proves to be a good alternative to dyadic decompositions.

Keywords: *subdivision surfaces, normal control, mesh deformation, N-adic scheme.*

1 Introduction

Manipulating efficiently and intuitively triangles meshes is a challenging problem for special effects and animation. Mesh models may be either created from scratch in an interactive modeling environment or obtained from digitized models. In general, those models are further edited. For instance, the surface may be smoothed, deformed, and details or extra features may be added. Subdivision surfaces have proved to be an attractive model for creating complex meshes.

Subdivision surfaces have been in the highlights of the computer graphics community for the past few years. One of the greatest advantage of subdivision algorithms is that they produce smooth surfaces from an arbitrary initial control mesh. Moreover, subdivision surfaces may be structured into a hierarchy, enabling the management of level of detail.

A number of subdivision schemes have been proposed so far and may be sorted into two categories. Approximation techniques move all the vertices of the mesh, whereas interpolation techniques keep the vertices of the control mesh unchanged. Approximation techniques have been most extensively studied as they tend to produce better shapes and are more flexible. Those methods do not provide a tight control of the surface, which is often crucial in surface design and character animation. In contrast, interpolation techniques preserve the vertices of the control mesh, but the fairness of the surface is more difficult to control.

In general, deformations are performed by moving control points. Local multiresolution deformations are achieved by editing the control mesh at different resolution levels [7, 21, 14]. A very attractive modeling feature is the ability to prescribe the normals of a mesh at given vertices using rotations. Such a control is very interesting, since it provides a complete set of tools in a context of field controlled shape deformation [12]. Still, performing normal control for subdivision surfaces remains a difficult challenge. Several deformations methods using normal control have been proposed, however most rely on approximation schemes [8, 2, 15].

In this paper, we present simple and efficient local deformation methods based on a new N-adic interpolating subdivision scheme. Control is performed with rotations by moving the normals either around the vertices of the control mesh, or around the normal axis so that the surface should be locally twisted. In contrast with dyadic schemes, we use a more generic N-adic decomposition of the parameter space which results in a better shape definition. We eventually compare deformations produced by our N-adic scheme and the standard dyadic method.

2 Previous work

Several authors have proposed techniques for editing and rendering subdivision surfaces efficiently. Most editing tools focus on the displacement of the vertices of the control mesh and try to optimize the internal representation of the surface with level of details.

Zorin et al. [21] have proposed an approach based on signal processing techniques. The surface is split into a set of hierarchical meshes that represent the increasing level of detail added by each finer mesh. The process is scalable, it may be performed automatically and enables local modifications on the surface. Although the Loop scheme [11] has been tested, it can be adapted to other schemes. Mandal et al. [14] have proposed a scheme for dynamic manipulation of the limit surface created with the modified Butterfly scheme [6, 19] using physically based force controls. The control is provided by tracking vertices at various levels of the subdivision. Although accurate, the surface deformation remains global and computationally expensive. Therefore, it is dedicated to simulation rather than interactive animation.

Several other methods deal with the deformation of the surface by modifying the normals at the vertices of the control mesh. Those methods are restricted to non-interpolating approximation schemes however.

Nasri [15] has proposed a simple yet efficient technique for the Doo-Sabin [4] scheme. The first subdivision step aims at creating new vertices that are relocated to adapt to a given normal. The Doo-Sabin specific corner cutting algorithm locates new vertices in the vicinity of initial control points. Further subdivision steps simply smooth the mesh. Halstead et al. [8] has improved the Catmull-Clark [3] scheme in order to directly generate the limit surface. Conditions for interpolating normals are given. Unfortunately, using normals to control the limit surface dramatically increases the resolution cost. Biermann [2] has proposed another method for approximating various schemes such as Loop or Catmull-Clark. This is achieved by adding a correction step after each classical subdivision step. Resulting points from this correction step are oriented relatively to the normal.

Some authors have proposed other iterative and easily deformable reconstruction methods.

Volino [17] uses blended spheres to perform a N-adic tessellation of triangles. Points and normals from the initial mesh are directly involved in its construction. Vlachos [18] uses a N-adic tessellation to perform his PN Triangles (or N-patches). Primitives used here for reconstruction are polynomials, based on a more current Bézier patch model. Its construction uses points and normals as well. This method has been proposed with a view to being implemented in the graphics hardware.

In our paper, we propose a N-adic interpolating subdivision scheme which enables us to perform intuitive one-step deformations based on a normal control. Section 4 describes the N-adic reconstruction process whereas section 5 describes the different deformation tools.

3 Background

In this section, we present the notations and some results that are useful in the remainder of the paper.

Subdivision surfaces are addressed from both a topological and geometrical point of view. The topology defines the structure of the mesh with vertices, edges and faces, whereas the geometrical aspect purposes the geometric representation of the vertices in \mathbb{R}^3 . When designing topological vertices, we use small letters whereas capital letters denote their geometric coordinates.

During the subdivision process, new vertices are inserted into the topology. Most subdivision schemes make use of a dyadic decomposition. Edges are split in two by inserting a new vertex in the middle of the edge, hence, triangles are subdivided in four. The geometric coordinates of the new vertices, referred to as elevation, are evaluated with a given transformation associated to the vertices of the mesh (Figure 1).

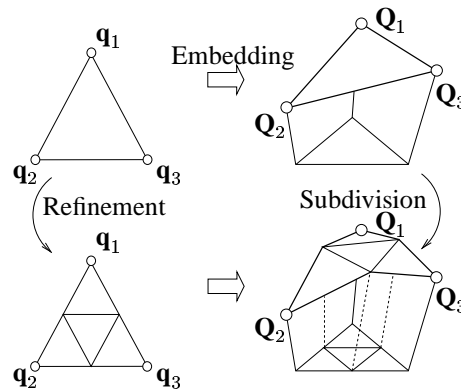


Figure 1. Subdivision process: new vertices are inserted into the topology (left) so that each old triangle (upper row) may be divided into four (bottom row). Coordinates are obtained after embedding the mesh into \mathbb{R}^3 (right).

The transformation may be characterized by masks that define the elevation of each new vertex as a barycentric combination of the coordinates of neighboring vertices. Masks depend on the local topology around each old vertex. They are represented with stencils that take into account the valence of the vertex, which is the number of adjacent vertices. In our work, we start from the modified

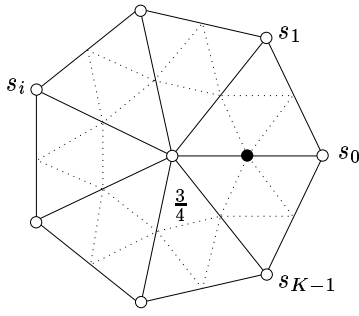


Figure 2. Modified Butterfly mask for an extraordinary vertex. Vertices involved in the computations for the black vertex belong to the 1-neighborhood of the central vertex.

1-neighborhood Butterfly masks described in [20]. Figure 2 illustrates the associated stencil applied near a vertex with valence K . Coefficients used for the barycentric combination are as follows, if $K \geq 5$ then:

$$s_i = \frac{1}{K} \left(\frac{1}{4} + \cos \frac{2i\pi}{K} + \frac{1}{2} \cos \frac{4i\pi}{K} \right) \quad (1)$$

If $K = 4$ we have:

$$s_0 = \frac{3}{8}, s_1 = s_3 = 0, s_2 = -\frac{1}{8} \quad (2)$$

Eventually, $K = 3$ yields:

$$s_0 = \frac{5}{12}, s_1 = s_2 = -\frac{1}{12} \quad (3)$$

These coefficients are chosen so that the elevation of the new vertices should reproduce polynomial surfaces. Let \mathbf{q}_1 and \mathbf{q}_2 denote the end vertices of an edge. For the modified irregular Butterfly scheme the elevation of the midpoint will be a combination of the two elevations B_1 and B_2 produced by the masks associated with \mathbf{q}_1 and \mathbf{q}_2 (see Figure 3).

In the next section we extend this edge decomposition to the insertion of a new vertex m at any location of a triangle $\Delta \mathbf{q}_1 \mathbf{q}_2 \mathbf{q}_3$. The related elevation denoted as $B(m)$, is defined on $\Delta \mathbf{q}_1 \mathbf{q}_2 \mathbf{q}_3$ to be split as follows, as several basic contributions $B_{\mathbf{q}_i}(m)$. In this paper, we restrict the support of $B_{\mathbf{q}_i}$ to the 1-neighborhood of the vertices \mathbf{q}_i . Therefore, only the three elevation functions $B_{\mathbf{q}_1}$, $B_{\mathbf{q}_2}$ and $B_{\mathbf{q}_3}$ participate to the final computation as shown in equation (4).

$$\forall m \in \Delta \mathbf{q}_1 \mathbf{q}_2 \mathbf{q}_3 \begin{cases} B(m) = \sum_{i \in \{1,2,3\}} \alpha_i B_{\mathbf{q}_i}(m) \\ \sum_{i \in \{1,2,3\}} \alpha_i = 1 \end{cases} \quad (4)$$

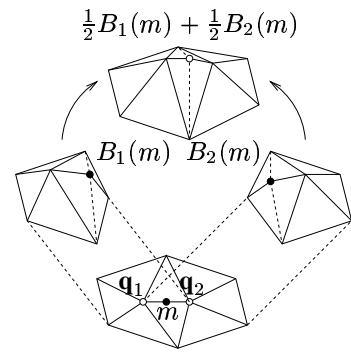


Figure 3. Inserting a new vertex between two vertices with the modified Butterfly scheme. The final elevation of vertex m is a combination of the elevations at vertices \mathbf{q}_1 and \mathbf{q}_2 .

4 N-adic surface reconstruction

In this section, we describe the subdivision scheme we designed to deform the mesh with normal control. The algorithm relies on the computation of the elevation described in the previous section. Let us recall that the elevation at a given point m is obtained by combining the local elevation functions $B_{\mathbf{q}_1}$, $B_{\mathbf{q}_2}$ and $B_{\mathbf{q}_3}$ centered at the 1-neighborhood of the vertices of the triangle $\Delta \mathbf{q}_1 \mathbf{q}_2 \mathbf{q}_3$.

4.1 Computation of a single elevation function

Let \mathbf{q} a vertex of the mesh with valence K . We operate on the 1-neighborhood of \mathbf{q} , which is defined by a K -sided polygon with vertices $\mathbf{q}_0, \dots, \mathbf{q}_{K-1}$. We aim at inserting a new vertex m in the 1-neighborhood of \mathbf{q} and computing its corresponding elevation $B_{\mathbf{q}}(m)$ as a function of existing vertices $\mathbf{q}, \mathbf{q}_0, \dots, \mathbf{q}_{K-1}$. The position of m in the local configuration will be described with its polar coordinates (ρ, θ) (see Figure 4).

To perform this step, we rely on the eigenanalysis developed by Zorin [20] which was first proposed in [1] and [5]. This scheme aims at reproducing polynomials that characterize the local geometry for specific vertices. It may be generalized to $m(\rho, \theta)$ over the first triangle. Formulas may be derived for other neighboring triangles by circular permutations. The elevation function $B_{\mathbf{q}}(m)$ may be written as a barycentric combination of the coordinates $\mathbf{Q}, \mathbf{Q}_0, \dots, \mathbf{Q}_{K-1}$.

$$B_{\mathbf{q}}(m) = s_{\mathbf{q}} \mathbf{Q} + \sum_{i=0}^{K-1} s_i^{(\rho, \theta)} \mathbf{Q}_i \quad (5)$$

Equation (5) may be compacted into a specific mask parametrized by ρ and θ (see Figure 4). The coefficients

$s_i^{(\rho, \theta)}$ depend on the value of the valence K :

$$s_i^{(\rho, \theta)} = \frac{\rho^2}{K} + 2 \frac{\rho}{K} \cos\left(\frac{2i\pi}{K} - \theta\right) + 2 \epsilon(K) \frac{\rho}{K} \cos\left(\frac{4i\pi}{K} - 2\theta\right) \quad (6)$$

where $\epsilon(K) = 1$ if $K \geq 5$, $\epsilon(K) = \frac{1}{2}$ if $K = 4$ and $\epsilon(K) = 0$ if $K = 3$. The coefficient $s_{\mathbf{q}}$ is equal to $1 - \rho^2$ whatever K may be.

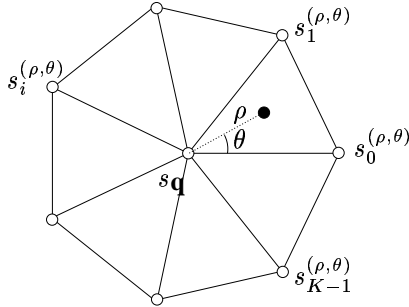


Figure 4. Subdivision mask for a vertex (ρ, θ) .

It is worth noticing that for $\rho = 0$, we have the interpolating condition $B_{\mathbf{q}}(\mathbf{q}) = \mathbf{Q}$. When $\theta = 0$, the new vertex is located on the edge. Therefore, if $\rho = 1/2$, we obtain the modified Butterfly mask for extraordinary vertices [20] when the new vertex is inserted at the middle of the edge. If $\rho = 1/3$, we obtain the coefficient computed by Labsik [10] that characterizes the elevation for the $\sqrt{3}$ -subdivision interpolation scheme.

4.2 N-adic scheme

Creating a N-adic scheme requires that each triangle $\Delta \mathbf{q}_1 \mathbf{q}_2 \mathbf{q}_3$ should be split into N^2 triangles. Thus, we need to define the elevation for each new vertex m_i as a function of $B_{\mathbf{q}_1}$, $B_{\mathbf{q}_2}$ and $B_{\mathbf{q}_3}$. The final elevation will be defined using the barycentric combination described in formula (4). In our implementation, the coefficients $(\alpha_1, \alpha_2, \alpha_3)$ are defined as the barycentric coordinates of the vertex m_i in the triangle. Those coordinates define the adequate coefficients (ρ_i, θ_i) for each single elevation function $B_{\mathbf{q}_i}$ as well.

This scheme provides a reconstruction method that creates coherent shapes. Figure 5 illustrates the smoothing of a werewolf like character after several triadic subdivision steps using our scheme. The N-adic decomposition provides a general framework for surface evaluation. Our deformation method strongly relies on the sampling properties of the N-adic scheme.

5 Surface deformation

In this section, we present our algorithm that controls the orientation of the subdivision surface in the neighbor-

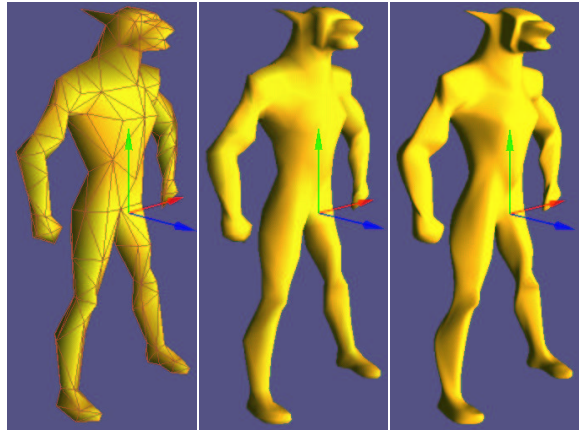


Figure 5. Recursive subdivision for a triadic scheme. From left to right: original mesh, after one triadic step, after 3 triadic steps.

hood of the vertices of the control mesh. First we assume that a normal denoted as $N_{\mathbf{q}}^0$ is evaluated at each vertex \mathbf{q} of the control mesh. This normal is often provided with the mesh for rendering purposes. We deform the surface in the neighborhood of each involved vertex \mathbf{q} by modifying the orientation of the single elevation function $B_{\mathbf{q}}$ associated to this vertex. This modification is controlled with different rotations parametrized by several action tools on the normal. Either the rotation changes the normal direction (normal rotation), or uses its axis as a rotation axis (twist). The final elevations of the new vertices are obtained by applying those rotations on the previous elevations.

5.1 Normal rotation based deformations

The first deformation tool aims at re-orienting the new normals that result from the subdivision process. The resulting configuration must be aligned with the normal $N_{\mathbf{q}}$ specified by the user. This orientation is characterized by a unique rotation $R_{\mathbf{q}}$ centered at \mathbf{q} transforming $N_{\mathbf{q}}^0$ into $N_{\mathbf{q}}$. $R_{\mathbf{q}}$ is then applied to the points to be computed by the function $B_{\mathbf{q}}$ (see Figure 6).

This control of the normals provides a simple and intuitive control on the shape deformation at the vertices of the control mesh. Deformations are performed after the first subdivision step. The next steps are applied without any deformation, for smoothing purposes.

Figure 7 illustrates one local deformation on a torus. Figure 8 shows a cube deformed by perturbing the normals at the four upper corners.

Figure 9 shows two examples of complex deformations. Left image represents a local deformation, whereas right image shows a model where all normals have been per-

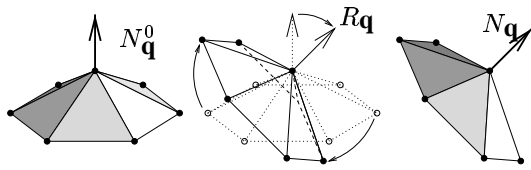


Figure 6. Local rotation at a vertex \mathbf{q} of the control mesh: after computing $B_{\mathbf{q}}$, each vertex inserted in the 1-neighborhood of \mathbf{q} is rotated using $R_{\mathbf{q}}$ that transforms $N_{\mathbf{q}}^0$ into $N_{\mathbf{q}}$.

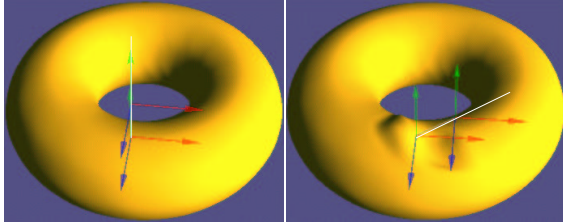


Figure 7. Perturbing one normal on a torus.

turbed. The process allows us to complete the design of local curvatures of objects only by moving the normals of its mesh. Figure 10 shows the enhanced features on the head of a werewolf character. Left image is obtained without normal deformation, whereas right image has been created by perturbing the control normals. The curved ear has been modeled by turning down the normal at its end. Modifying the normals of the neck produces a new curved shape below the chin and on its nape. The mouth and the face were modified as well.

5.2 Twist based deformations

Another basic deformation tool consists in twisting the surface around a vertex \mathbf{q} of the control mesh. The points of the local configuration turn around the normal axis $N_{\mathbf{q}}^0$ of the surface.

The user adds an extra angular information θ to the normal direction, to control the amplitude of the deformation. Therefore, those two parameters define a single extra rotation $T_{\mathbf{q}}$, which characterizes the twist entirely. Like the normal rotation operator $R_{\mathbf{q}}$, the twisting operator $T_{\mathbf{q}}$ is applied to the points obtained with the elevation function $B_{\mathbf{q}}$ (see Figure 11).

Twisting proves to be a very intuitive method that offers extra degree of freedom for quick shape deformation. Figure 12 shows the influence of the angular parameter θ on the deformation. The method still does not preserve the volume of the object for strong deformations. Figure 13 shows an example of twisting perturbations on an icosahedron.

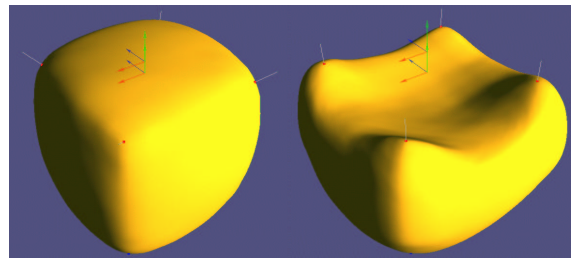


Figure 8. Deforming one face of a cube.

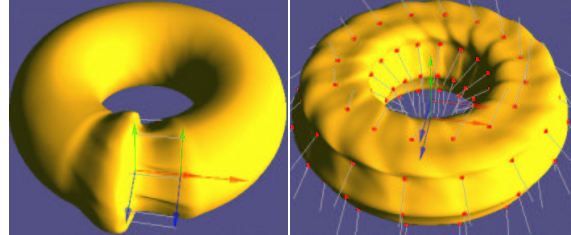


Figure 9. Several deformations applied on the previous torus.

5.3 Combining normal rotation and twist

Normal rotation and twist can be easily combined as a unique extended operator. During the process, twists will be actually applied to the previously rotated surface.

For a vertex \mathbf{q} , the points resulting from the elevation function $B_{\mathbf{q}}$ are first moved using the rotation $R_{\mathbf{q}}$ which aligns the configuration along the specified normal $N_{\mathbf{q}}$. To this modified elevation function $B'_{\mathbf{q}} = R_{\mathbf{q}}(B_{\mathbf{q}})$ we apply the twist $T_{\mathbf{q}}$ with the new normal $N_{\mathbf{q}}$ to the new vertices. We eventually obtain the final configuration (see Figure 14).

Thus a single transform $P_{\mathbf{q}} = T_{\mathbf{q}} \circ R_{\mathbf{q}}$ defines the deformation characterized by a normal direction and an angle of twist that are specified by the user. Experiments show that this order of composition proves to be crucial to have an intuitive deformation tool. This general tool may be used to generate a vast variety of shapes given an initial coarse mesh (see Figure 15).

6 N-adic vs. dyadic schemes

This section compares the properties of our N -adic subdivision scheme with a standard dyadic scheme. Let us point out the following remarks.

The very first steps of subdivision aims at defining the global shape of the object, whereas the last steps only smooth the object and have very little influence over the shape.

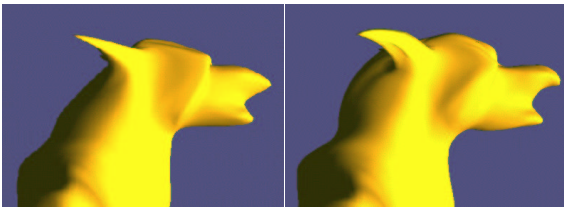


Figure 10. Redesigning a werewolf character.

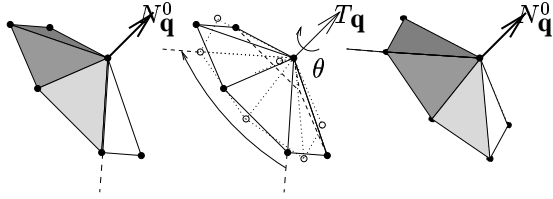


Figure 11. Twist around a vertex \mathbf{q} of the control mesh: each vertex inserted in the 1-neighborhood of \mathbf{q} is rotated using $T_{\mathbf{q}}$. The new local configuration is finally turned around $N_{\mathbf{q}}^0$, with an angle θ .

Dyadic subdivision schemes are popular for final smoothing, but their recursive structure increases the number of inserted vertices dramatically. The resulting lack of scalability leads to a poor control over the geometric definition of the shape. Therefore, those algorithms prove to be ill-suited for deformation purposes. Figure 16 illustrates our deformation method with the standard dyadic scheme: the lack of definition for the first sub-mesh results in some defaults in the final shape.

A scalable definition of the first steps is crucial for a quick design or deformation purposes. To fulfill this goal, iterative schemes like the Spherigon [17] or N-patches [18] are very interesting as they generate adaptive meshes. Their natural N-adic one-step decomposition makes them very attractive, since they enable to add extra vertices of the new sub-mesh. Those extra vertices provide extra informa-

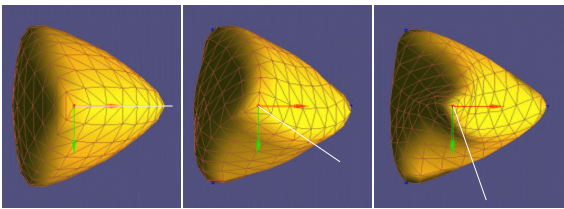


Figure 12. Twisting a tetrahedron for different values of θ . From left to right: without deformation, 30 degrees, 60 degrees.

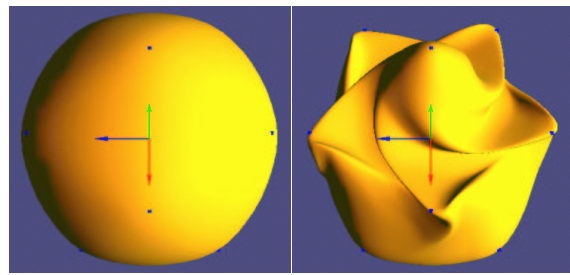


Figure 13. Perturbation on an icosahedron, obtained by twisting the upper control vertices (left: without deformation; right: after twist).

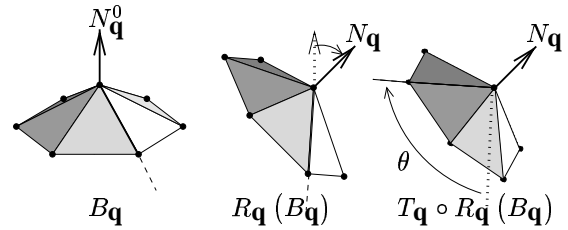


Figure 14. Combination of a rotate and a twist around a vertex \mathbf{q} of the control mesh: each vertex inserted with $B_{\mathbf{q}}$ in the 1-neighborhood of \mathbf{q} is first rotated using $R_{\mathbf{q}}$. The new local configuration is then twisted around the new normal $N_{\mathbf{q}}$, with $T_{\mathbf{q}}$.

tions which results in a better sampling for a one-step shape control. Therefore, such schemes are often used for fast hardware rendering. Unfortunately, they are ill-suited for smoothing. Stam [16] proposed a method that combines the scalability of an iterative scheme and preserves the recursive properties of subdivision schemes. The method relies on an N-adic subdivision algorithm which reproduces Catmull-Clark surfaces. Maillot [13] enhanced the previous scheme with some extra rules and obtains a better shape control for rendering. This method produces a larger variety of meshes for a scalable rendering process.

Our N-adic scheme actually aims at following the same ideas for butterfly surfaces. The first N-adic step generates a more accurate shape for our deformation methods, while the last steps perform some smoothing. Figure 17 shows both dyadic and triadic subdivisions we used in practice to perform our experiments, and illustrates this improved shape definition for a tetrahedron. This sampling accuracy improves the quality of the mesh definition after the first subdivision step. Figure 18 illustrates the deformations of a face of a cube using the dyadic and the triadic schemes. The triadic scheme produces a smoother surface, and creates a deeper carving than the dyadic scheme. The N-adic version of the modified butterfly algorithm has one drawback how-

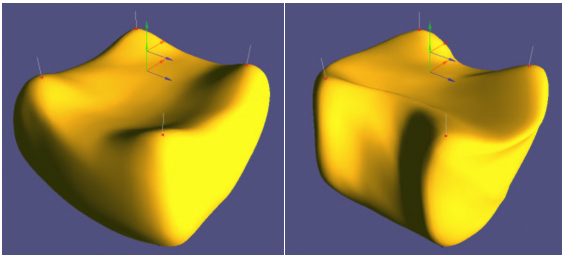


Figure 15. Combinations of rotate and twists on a cube on its upper control points. Left: deformations after single rotations; right: deformations after a twist add-on.

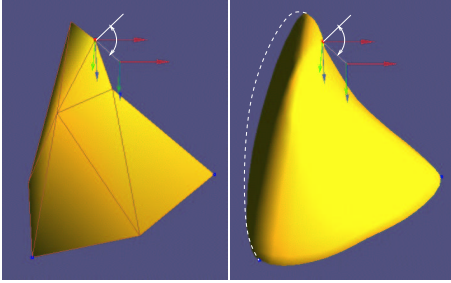


Figure 16. Deformation of a tetrahedric mesh after one step of dyadic decomposition (left). The poor number of new vertices for the mesh definition results in a bad final shape structure or a drift between the true normal to the surface and the modified one (right).

ever for deformations applied on high level meshes. Extra inserted points can alter the quality of the final shape at limit of deformed areas and create creases or cusps. Experiments show that these artefacts increase for oversampled meshes. In these cases, the limit surface may significantly differ from the original Butterfly surface. As a conclusion, the N-adic subdivision scheme lends itself for controlling the first subdivision steps. Standard dyadic subdivision methods are more suitable for final smoothing.

7 Conclusion and future work

We have presented an interpolating subdivision scheme based on an N-adic decomposition of the parameter space. This approach creates a general surface framework for subdivision surface deformation.

We have implemented the triadic case so as to locally deform the surface according to a modification of the normals at the vertices of the control mesh. Several tools allow us to control the different parameters of the normal configura-

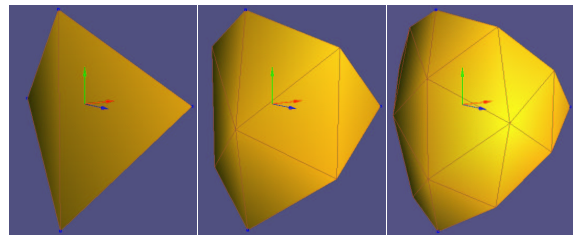


Figure 17. Structure of a dyadic (middle) and a triadic decomposition (right) after one subdivision step. The original object is displayed on the left.

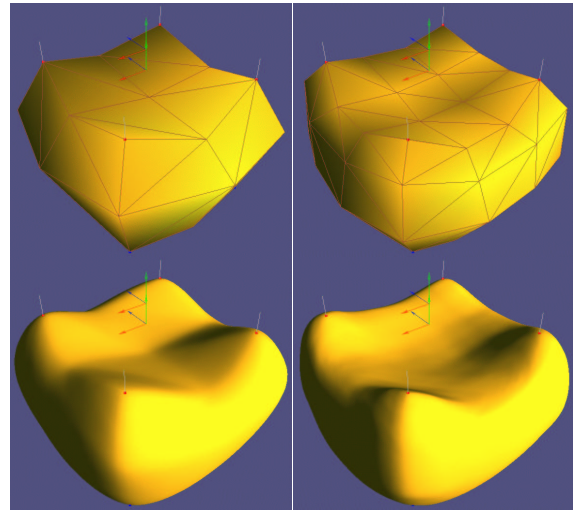


Figure 18. Deformation of a face of a cube for the dyadic (left) and the triadic case (right). Upper row: structure of the mesh after one deformation. Bottom row: final shaded shapes after some smoothing steps.

tion: normal rotation which changes the normal direction and twist which rotates the surface around its axis.

We have compared both dyadic and N-adic schemes. Our N-adic decomposition provides a better one-step shape control when invoking the deformations, whereas the dyadic decomposition method produces smoother surfaces.

Several topics need further research. A major drawback of most subdivision methods is the recursive form of the algorithms. All subdivision levels need to be computed, even for triangles that will be discarded in the rendering pipeline. The iterative aspect of our method enables us to avoid subdivision wherever unnecessary. It could be applied to perform view dependent subdivision.

Our method currently deforms the surface by modifying the position of the vertices of the first level subdivision step. The extra steps are used for smoothing purposes. We plan

to extend the deformation to other levels so as to emphasize deformations at higher levels of detail. In the near future we plan to implement multilevel dyadic deformations and compare them to the current results.

In our shape generation method, the edges are uniformly subdivided into N sub-edges of equal length for each triangle. A smarter repartition of the new vertices would create a new mesh that would fit the surface better. We could migrate vertices towards regions of high curvature, leaving flat areas with fewer points. In this scope, the surface could be computed by our elevation function at arbitrary vertex position.

References

- [1] A.A. Ball and D.J.T. Storry. Analysis of the behaviour of Recursive Subdivision Surfaces Near Extraordinary Points. *ACM Transactions on Graphics*. **7**(2) : 83-102, 1988.
- [2] H. Biermann, A. Levin and D. Zorin. Piecewise smooth subdivision surfaces with normal control. *Computer Graphics (SIGGRAPH 2000 proceedings)*. 113-120, July 2000.
- [3] E. Catmull and J. Clark. Recursively generated B-Spline Surfaces on arbitrary topological meshes. *Computer Aided Design*. **10**(6) : 350-355, 1978.
- [4] D.Doo and M. Sabin. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. *Computer Aided Design (Interactive Techniques proceedings)*. 157-175, 1978.
- [5] D.Doo and M. Sabin. Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, **10**(6) : 356-360, 1978.
- [6] N. Dyn, D. Levin and J. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, **9**(2) : 160-169, April 1990.
- [7] D. Forsey and R. Bartels. Hierarchical B-Spline refinement. *Computer Graphics (SIGGRAPH 1988 proceedings)*, **22**(4) : 205-212 August 1988.
- [8] M. Halstead, M. Kaas and T. DeRose. Efficient, fair interpolation, using Catmull-Clark surfaces. *Computer Graphics (SIGGRAPH 1993 proceedings)*, 35-44, 1993.
- [9] L. Kobbelt. $\sqrt{3}$ -Subdivision. *Computer Graphics (SIGGRAPH 2000 proceedings)*, 103-112, July 2000.
- [10] U. Labsik and G. Greiner. Interpolatory $\sqrt{3}$ -Subdivision. *Computer Graphics Forum (Eurographics 2000)*, **19**(3) : 131-138, 2000.
- [11] C. Loop. Smooth subdivision surfaces based on triangles. *Master thesis, University of Utah, U.S.A*, August 1987.
- [12] J. Maillot. Real Time Approximation of Deformations using Rotations. *Computer Graphics Forum (Eurographics 2000)*, **19**(3) : 131-138, 2000.
- [13] J. Maillot and J. Stam. A Unified Subdivision Scheme for Polygonal Modeling. *Computer Graphics Forum (Eurographics 2001)*, **20**(3) : 471-479 , 2001.
- [14] C. Mandal, H. Qin and Baba C. Vermuri. Dynamic Modeling of Butterfly Subdivision Surfaces. *IEEE Transaction On Visualization And Computer graphics*, **19**(3) : 427-436, July-September 2000.
- [15] A.H. Nasri. Surface interpolation on irregular networks with normal conditions. *Computer Aided Geometrics Design*, **23**(6) : 405-410, 1991.
- [16] J. Stam. On Subdivision Schemes Generalizing Uniform B-Spline Surfaces of Arbitrary Degree. *Computer Aided Geometric Design*, **18** : 383-396, 2001.
- [17] P. Volino and N. Magenat Thalmann. The Spherigon: a simple polygon patch for smoothing quickly your polygonal meshes. *Computer Animation 98 proceedings*, 72-79, 1998.
- [18] A. Vlachos, J. Peters, C. Boyd and J.L. Mitchell. Curved PN Triangles. *ACM Symposium on Interactive 3D Graphics*, 159-166, March 2001.
- [19] D. Zorin, P. Schröder and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. *Computer Graphics (SIGGRAPH 1996 proceedings)*, 189-192, 1996.
- [20] D. Zorin, P. Schröder and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. *Technical Report, California Institute of Technology, U.S.A*, June 1996.
- [21] D. Zorin, P. Schröder and W. Sweldens. Interactive Multiresolution Mesh Editing. *Computer Graphics (SIGGRAPH 1997 proceedings)*, 159-168, 1997