Least Squares Subdivision Surfaces

Simon Boyé, Gael Guennebaud, Christophe Schlick

Pacific Graphics 2010
P. Alliez, K. Bala, and K. Zhou
(Guest Editors)

Volume 29 (2010), Number 7

Least Squares Subdivision Surfaces

S. Boyé, G. Guennebaud and C. Schlick

INRIA – LaBRI (Bordeaux University - CNRS)

Abstract
The usual approach to design subdivision schemes for curves and surfaces basically consists in combining proper rules for regular configurations, with some specific heuristics to handle extraordinary vertices. In this paper, we introduce an alternative approach, called Least Squares Subdivision Surfaces (LS^3), where the key idea is to iteratively project each vertex onto a local approximation of the current polygonal mesh. While the resulting procedure haves the same complexity as simpler subdivision schemes, our method offers much higher visual quality, especially in the vicinity of extraordinary vertices. Moreover, we show it can be easily generalized to support boundaries and creases. The fitting procedure allows for a local control of the surface from the normals, making LS^3 very well suited for interactive freeform modeling applications. We demonstrate our approach on diadic triangular and quadrangular refinement schemes, though it can be applied to any splitting strategies.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

During the last twenty years, subdivision surfaces have become increasingly popular as a powerful surface representation for interactive modeling applications [ZS00]. A subdivision surface takes as input a relatively coarse polygonal mesh with arbitrary (manifold) topology and produces a smooth surface through an iterative refinement process. This refinement naturally yields a multiresolution structure which is particularly useful to enable adaptive rendering or to efficiently edit 3D objects at different scales [ZSS97, Zor06]. Furthemore, subdivision surfaces offer a relatively high degree of flexibility to artists: arbitrary topology, sharp edges and corners with controllable sharpness [DKT98], etc. For all these reasons, subdivision surfaces are now a well established standard, especially in the game and movie industries.

So far, many different schemes have been proposed [CC78, DS78, Loo87, DLG90, Kob96, PR97, Kob00, CADS09]. In most cases, subdivision rules are designed to reproduce some given (box-) splines. However, such rules can be derived in the regular case only. Consequently, subdivision surfaces are also famous to exhibit severe artifacts around so called extraordinary vertices [SB02] for which no ideal subdivision rule can be derived. In particular, we can distinguish three combined effects: the polar artifact (contraction or dilatation around vertices of low or high valence

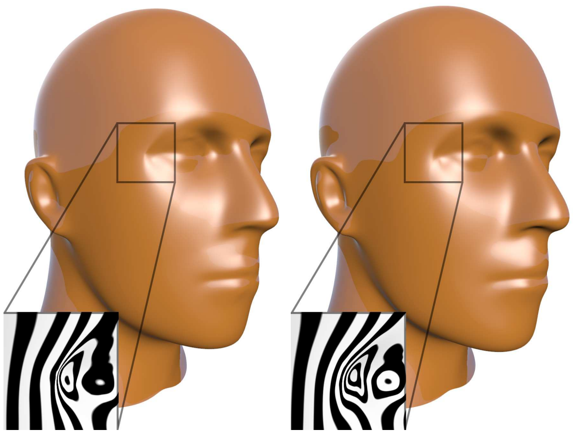
Figure 1: *Loop's subdivision surface (left) and our LS^3 method (right). Note the pinch in the shading in the vicinity of the extraordinary vertex lying at the eye corner.*

respectively), poor fairness (ripples), and the lack of curvature continuity.

During the last decade, many work has been focused on improving the behavior of subdivision surfaces around extraordinary vertices. The classical approach consists in tuning extraordinary point rules by analyzing the eigenstructure of the subdivision matrix [DS78, Rei96, ZS00, WW02, BK04]. While most approaches strive to satisfy bounded cur-

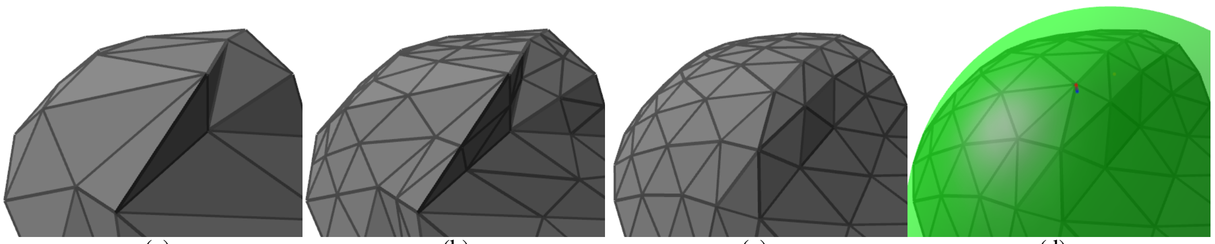

Figure 2: *Illustration of one LS^3 iteration. Starting from the current mesh (a), we first apply the splitting rule to obtain a denser mesh (b). Then the relaxation rule regularizes the mesh (c). The final position of each vertex (d) is obtained by projecting it onto its respective local least-squares approximation patch (shown in green).*

vature, it has been shown that such schemes are likely to decrease the surface fairness [KPR04] and increase the polar artifact [BK04] yielding surfaces of poor visual quality. Cashman et al. [CADS09] shown higher degree subdivision can improve the surface fairness at the cost of increased computation, increased shrinkage, and less local influence of the control points. Levin [Lev06] managed to guaranty C^2 continuity by carefully blending the limit surface in the vicinity of extraordinary vertices to a C^2 polynomial patch. This is a non stationary scheme which requires a special non trivial handling of such regions, and a few initial refinement iterations which already degrades the surface fairness. While focusing on C^2 continuity and/or bounded curvature makes sense for CAD applications, we are primarily interested in freeform modeling for game-like and movie applications where visual quality (fairness), simplicity and efficiency are more critical. These last two criteria become even more obvious with the introduction of a tessellation engine into graphic cards, since we can expect that subdivision surfaces will also become a standard of real-time rendering engines.

In this paper we propose a novel approach to design approximating subdivision surfaces called **Least Squares Subdivision Surfaces** (LS^3 for short). The key idea behind LS^3 is to explicitely decouple the *smoothing rule* of classical subdivision surfaces into a *relaxation rule* regularizing the mesh, and a parameterization-free *projection operator*. The later offers new possibilities to control the shape and smoothness of the surface regardless of the other steps of the subdivision. Central to our framework is the choice of the projection operator. In this paper, we focus on the use of local least-square approximations. At each step, each vertex is relaxed by centroid averaging and then it is projected onto a dynamically fitted surface patch locally approximating the old neighbor vertices. This is inspired from the Point Set Surfaces (PSS) literature [ABC*03] where surfaces with arbitrary smoothness can be defined by mean of local *Moving Least Squares* (MLS) approximations [Lev03]. Unlike our approach, PSS require strong sampling criteria on the input point clouds as they do not rely on any connectivity information.

In particular, we adapt the Algebraic Sphere Fitting (ASF) procedure developed by Guennebaud et al. [GGG08] which offers a good compromise between the approximation power and the performance. The key feature of ASF is the clever use of surface normals such that high stability is achieved with a very few number of input points. In other words, a first-ring neighborhood plus normals is almost equivalent to a 2-ring neighborhood, thus allowing us to generate surfaces of very high quality with very limited overhead. As shown by our experimental results, the new approach significantly reduces the artifacts of classical subdivision surfaces, while treating all vertices in a uniform manner. Owing to the non linearity of our scheme, we also developed numerical tools to observe the C^0, G^1, and G^2 behaviors where the subdivision scheme is seen as a black box.

Furthermore, the input normal vectors may be freely tweaked by the user, to offer additional fine control of the final surface shape. LS^3 also supports boundaries and sharp edges with controllable sharpness. Finally, LS^3 is generalizable to any splitting strategy, though we mainly focus here on primal diadic subdivison schemes such as Loop and Catmull-Clark schemes.

2. Least Squares Subdivision

Our method follows the same principle than classical subdivision surfaces: an initial polygonal mesh M^0 with arbitrary connectivity is iteratively refined by applying a subdivision operator to the currently refined mesh. This process generates a sequence of meshes $M^0, M^1, M^2, \ldots$ with increasing density which converges to a final limit surface. The subdivision operator is usually a two-step procedure: splitting and smoothing. In order to gain more flexibility, this last step is decloupled into a relaxation rule combined with a projection operator. This lead to the following three-step procedure illustrated in figure 2.

The splitting rule refines the mesh by inserting new vertices and creating new faces (fig. 2-b). To simplify the description, we only consider primal diadic refinements of triangular meshes, but the process can be easily generalized to other refinement schemes, such as quadrangular meshes, dual or $\sqrt{3}$ subdivisions.

The relaxation rule aims at producing meshes which are locally uniform by weighted affine combinations of the old

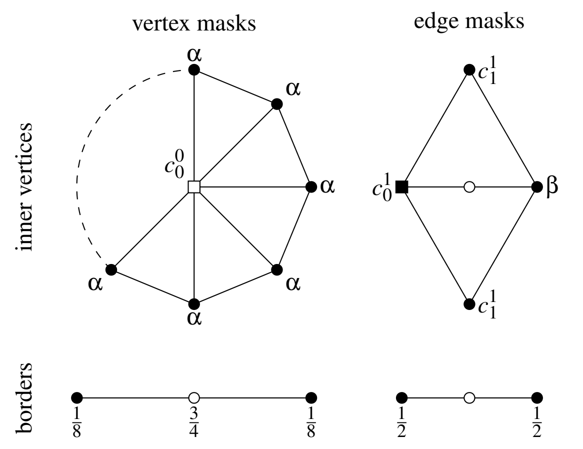

Figure 3: *Generic relaxation masks where the weights c_0^0, c_0^1, and c_1^1 depend on the valence n of the square vertex and are given in [BK04]-Table 2.* $\alpha = (1 - c_0^0)/n$ *and* $\beta = 1 - c_1^0 - 2c_1^1$.

vertices (fig. 2-c, section 3). In practice this step is similar to the smoothing step of classical subdivisions.

The projection operator aims at improving the surface fairness by projecting each relaxed vertex onto an algebraic surface locally approximating the old vertices in a weighted least-square sense. (fig. 2-d, section 4).

Since the splitting rule remains standard, in the next sections we will only focus on the two other steps of the algorithm, and we will show how boundaries and sharp edges can be properly handled in section 5. Section 6 presents numerical tools to analyze the continuity of our surface.

3. Relaxation rule

The purpose of the relaxation rule is to locally tend towards more uniform meshes that appears to be a necessary condition to generate surfaces of high visual quality. For instance, shading algorithms have a strong ability to depict mesh irregularities and exhibit odd behaviors in the case of, e.g., elongated triangles. Moreover, as we will discuss later, local uniformity is important to ensure a good behavior of our projection operator.

Ideally, the relaxation rule should move the vertices tangentially such that:

- the edges form smooth curves with smoothly varying edge lengths,
- the neighbors of each vertex are uniformly spread onto a given ellipsoid with smoothly varying shape,
- the initial shape of the mesh is preserved.

For regular vertices, these goals are already well satisfied using the smoothing masks of classical approximating subdivision schemes. For extraordinary vertices, however, they usually exhibit strong dilatation or contraction of the vertices in the case of high or small valence respectively. Fortunately, it has been shown that this *polar artifact* can be easily overcome by relaxing some C^2 necessary conditions [BK04]. Such rules are ideal in our case since the final shape of the surface will be given by the projection operator.

More precisely, this relaxation step boils down to classical subdivision. The intermediate position $\mathbf{q}_i^k$ of each vertex produced by the splitting rules is computed by weighted affine combinations of the old vertex positions $\mathbf{p}_j^{k-1}$ of the mesh M^{k-1}. The respective masks are shown in figure 3 where the weights c_0^0, c_0^1, and c_1^1 are appropriately set to reduce the polar artifact according to [BK04]- Table 2.

4. Projection operator

The relaxation step alone already produces relatively smooth surfaces most of the time, but with severe smoothness and fairness issues in the vicinity of extraordinary vertices. The role of the projection operator is to alleviate these issues while offering additional controls on the limit surface by mean of the input surface normals.

In a nutshell, our projection operator consists in projecting each vertex of position $\mathbf{q}_i^k$ coming from the relaxation rule, onto a simple surface patch that locally approximates the neighbor vertices of the previous mesh M^{k-1}. This yields the final positions $\mathbf{p}_i^k$ of the new mesh M^k. In the rest of this section, we will motivate our choice for a given local approximation method and discuss the details.

4.1. Local surface approximation

When it comes to local surface approximation, polynomials are an obvious choice for the corresponding basis functions. However, the minimization procedure and the accuracy of the approximation significantly vary according to the parameterization on which the polynomial surface is defined. For instance, a first possibility would be to compute a 2D local parameterization of the neighborhood and fit a bivariate $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ polynomial surface. However, it remains the difficulties of computing such a (local) parameterization in a consistent fashion across the different combinations, and to compute the projection onto it. A simpler possibility would be to use a planar parameterization by first fitting a reference plane and then fitting a bivariate polynomial height field ($\mathbb{R}^2 \rightarrow \mathbb{R}$) as it was done in former PSS definitions [ABC*03]. However, such an approach requires locally relatively flat data such that the plane fit is stable enough and that the neighborhood can be well represented as a height field [AK04b].

More recently, it has been shown that PSS can be advantageously defined by directly working in the natural cartesian parameterization by fitting algebraic (i.e., implicit) surfaces onto the sample points [AK04a, GG07]. More precisely, such approaches fit trivariate $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ implicit polynomials such that the *0-isosurface* $S = \{\mathbf{x} \in \mathbb{R}^3; s(\mathbf{x}) = 0\}$ is as close as possible to the input sample points. In order to avoid the trivial solution $s = 0$, additional regularization constraints have to be added. While it is possible to fit implicit planes [AA04], such a solution is far to be ideal in our case as it would suffer from the same aforementioned limitations of planar parameterizations. More recently, Guennebaud et al. introduced an efficient Algebraic Sphere Fitting (ASF) method [GGG08] to robustly approximate points equipped with normals by spheres. Their method naturally deals with planar regions, and thanks to the use of the normals, it requires only two input points to be well constrained. Therefore, minimal neighborhoods can be used that is especially important to avoid the need to loop through multiple neighborhood rings that is a tedious and expensive task.

In the next sections we briefly review ASF and show how it is adapted to our case.

4.2. Algebraic sphere fitting

An algebraic sphere is defined as the *0-isosurface* of a scalar field $s(\mathbf{x}) = [1, \mathbf{x}^T, \mathbf{x}^T\mathbf{x}]\mathbf{u}$ where $\mathbf{u} \in \mathbb{R}^{d+2}$ is a vector of parameters defining the shape of the sphere, and d is the dimension of the ambient space (in our case $d = 3$). This representation allows the sphere to naturally degenerate to a plane, which is required to approximate planar regions and to robustly move across inflexion points. The key idea of the ASF procedure [GGG08] is to first minimize for the constraint $\nabla s(\mathbf{p}_i) = \mathbf{n}_i$ stating that the gradient of s at each input constraint position $\mathbf{p}_i$ is as close as possible to the surface normal $\mathbf{n}_i$ attached to this point. This allows to solve for the $d+1$ coefficients u_1 to u_{d+1} of $\mathbf{u}$. The constant coefficient u_0 is obtained by minimizing the standard distance constraint ($s(\mathbf{p}_i) = 0$). This leads to a very fast fitting procedure with closed form formulas:

$$\begin{aligned} u_{d+1} &= \frac{1}{2}\frac{\sum w_i \mathbf{p}_i^T \mathbf{n}_i \;-\; \tilde{\mathbf{p}}^T \sum w_i \mathbf{n}_i}{\sum w_i \mathbf{p}_i^T \mathbf{p}_i \;-\; \tilde{\mathbf{p}}^T \sum w_i \mathbf{p}_i} \\ \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix} &= \sum w_i \mathbf{n}_i - 2u_{d+1}\tilde{\mathbf{p}} \qquad (1) \\ u_0 &= -[u_1 \dots u_d]\tilde{\mathbf{p}} \;-\; u_{d+1} \sum w_i \mathbf{p}_i^T \mathbf{p}_i \end{aligned}$$

where w_i are the weights associated to the constraint points $\mathbf{p}_i$ (such that $\sum w_i = 1$), and $\tilde{\mathbf{p}}$ is the weighted centroid $\sum w_i \mathbf{p}_i$. We refer to [GGG08] for more details.

The projection of an arbitrary point onto an algebraic sphere is performed by converting it to an explicit sphere or plane if $|u_{d+1}| > \varepsilon$ or $u_{d+1} = 0$ respectively. If u_{d+1} is close to zero, then a few newton iterations are required for high stability [GGG08].

4.3. Topological weighting

Most sophisticated PSS methods use for the weights w_i ellipsoid weight functions allowing to handle some limited anisotropy in the input data [AA06]. In our case, in order to avoid any sampling issue and to make sure the fitted sphere matchs well our input, these weights should follow the mesh structure as do the relaxation masks. Moreover, let us remark that the weights of the relaxation masks are barycentric coordinates of the point $\mathbf{q}_i^k$ for the old neighbor vertices $\mathbf{p}_j^{k-1}$. Therefore the most natural and obvious choice for the fitting weights w_i is to take the exact same masks and weights than the ones used at the relaxation steps.

This choice is also motivated by the following remark. To ensure a smooth transition of the fitted spheres, and thus, a high quality surface, such topological weights should tend to meshless (i.e., ellipsoidal) weights between the point $\mathbf{q}_i^k$ and the old neighbor vertices $\mathbf{p}_j^{k-1}$. This goal can only be reached by the combination of the relaxation rule and an appropriate weighting scheme for the fitting of the spheres.

4.4. Normal update

Finally, the normal $\mathbf{n}_i^k$ of each new vertex $\mathbf{p}_i^k$ is set to the normal of the locally fitted sphere s at the projection point, i.e., $\mathbf{n}_i^k = \nabla s(\mathbf{p}_i^k)$. These normals will be used at the next iteration to fit the spheres. We emphasize that these normal vectors does not necessarily match well the actual surface, and for shading, more accurate normals have to be computed from the mesh itself. On the other hand, using such normal vectors instead of recomputing them from scratch at each iteration has several advantages: it is faster, it allows to better propagate user specified normals through the subdivision levels, and it yields smoother surfaces since our projection step behaves like a smoothing operator of the normal vectors.

5. Boundaries and Sharp edges

As with any subdivision method, boundaries and sharp edges require some special treatments. For borders we propose to simply employ the exact same three step procedure than for the rest of the surface using the standard boundary masks and weights for both the relaxation and projection steps. The relaxation step ensures the necessary tangential smoothing of the curve, while the projection will make sure the curve is consistently "aligned" with the rest of the surface (see figure 4 for an illustration of the the effect of this projection). In the case of primal diadic subdivision, these masks implement cubic spline subdivision and they are recalled in figure 3. Let us also recall that the fitting procedure we adopted works well with only two vertices, so these very small masks are not an issue at all.

Sharp edges are usually handled by treating them as two separated boundaries. This is no longer possible with our

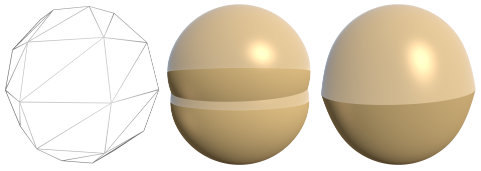
Figure 4: *A coarse sphere model with a sharp edge (left), subdivided using cubic spline rules (middle), and our LS^3 specific rules (right).*

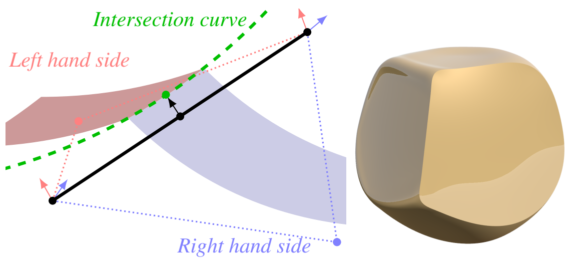

Figure 5: *Left, a sharp edge (bold line) is shared by two different smooth surfaces. During the refinement, the extremity vertices of the edge have different surface normals yielding to two different local spherical approximations. The new vertex (at the center of the edge) is projected onto the intersection curve of these two spheres. Right, illustration of sharp and semi-sharp creases.*

method since we are taking into account the surface normals and both sides are very unlikely to converge to the same curve. As illustrated in figure 5-left, our solution consists in fitting an algebraic sphere on each side as for boundaries, and then project the common relaxed point onto their intersection curve. The sharpness can be easily controlled by repeating a fixed number of sharp iterations followed by smooth iterations as explained in [DKT98], and shown in figure 5-right.

6. Numerical Analysis

Subdivision surfaces are usually analyzed through the eigen-structure of their respective subdivision matrix (e.g., [ZS00]). Unfortunately, owing to the projection step, in our case the new position of a vertex cannot be expressed as a linear combination of the old vertex positions, and therefore the subdivision matrix cannot be formed. This makes the theoretical analysis of our scheme rather difficult.

In the context of subdivision curves, a well established method to analyze non linear schemes is to numerically compute the Hölder regularity of several limit curves and retains the worst case [Kui98, MDL05]. Even though this is limited to curves, it still allows us to get some initial insights on the effect of our projection operator. Using the methodology described in [Kui98] and applying our LS^3 approach to planar 2D curves using the weighting masks of cubic spline subdivision (figure 3-bottom), we found that our projection step does not degrade the Hölder regularity which is in both cases greater than two.

The case of surfaces is more difficult, but also more interesting as it contains extraordinary vertices. To this end, we developed a numerical parameterization-free method to analyze the C^0, G^1, and G^2 behaviors of any subdivision surface. In a nutshell, we generate several 2-ring control meshes around a central vertex **p** of a given valence, subdivide until numerical convergence, and record some rates of convergence for each continuity degree. In order to be able to perform a lot of iterations, at each iteration we safely remove the two most exterior rings without affecting the result. Note that in all cases, the iterations always converged.

C^0 behavior is analyzed by recording at each iteration k the contraction factor ρ_k as the maximal length ratio of the edges around **p** between two successive iterations:

$$\rho_k = \max_j \frac{\|\mathbf{p}_j^k - \mathbf{p}^k\|}{\|\mathbf{p}_j^{k-1} - \mathbf{p}^{k-1}\|} \tag{2}$$

where $\mathbf{p}_j^k$ are the neighbor positions of the central vertex of position $\mathbf{p}^k$ at the level k. ρ_k must be strictly lesser than one and, for a diadic scheme, ρ_k should ideally be equal to 0.5.

G^1 behavior is equivalent to analyze the convergence rate of the neighborhood of **p** towards a planar configuration. To this end, we propose to analyze the eigen-structure of the covariance matrix $\mathbf{D}^k = \sum_j (\mathbf{p}^k - \mathbf{p}_j^k)^T (\mathbf{p}^k - \mathbf{p}_j^k)$. Let λ_i^k be the three eigenvalues of $\mathbf{D}^k$ sorted such that $\lambda_0^k < \lambda_1^k < \lambda_2^k$. Since the magnitude of each eigenvalue varies quadratically with respect to the scale of the input data, we can estimate the contraction factor α_k towards a planar configuration as:

$$\alpha_k = \left(\frac{\lambda_0^k / \lambda_1^k}{\lambda_0^{k-1} / \lambda_1^{k-1}} \right)^{\frac{1}{2}} \tag{3}$$

Again, α_k must be strictly lesser than one and, for a diadic scheme, α_k should ideally be equal to 0.5.

Assuming G^1 continuity, we can assign to the vertex $\mathbf{p}^k$ (resp. its neighbors $\mathbf{p}_j^k$) a normal $\mathbf{n}^k$ (resp. $\mathbf{n}_j^k$) as the eigenvector of the covariance matrix $\mathbf{D}^k$ (resp. $\mathbf{D}_i^k$) corresponding to the smallest eigenvalue. Then, G^2 behavior can be observed by comparing the convergence speed of the neighbor normals $\mathbf{n}_j^k$ to the normal $\mathbf{n}^k$ of the central central vertex, to the convergence speed of the vertex positions:

$$\beta_k = \frac{\operatorname{arclength}(\mathbf{n}^k, \mathbf{n}_j^k)\|\mathbf{p}_j^{k-1} - \mathbf{p}^{k-1}\|}{\operatorname{arclength}(\mathbf{n}^{k-1}, \mathbf{n}_j^{k-1})\|\mathbf{p}_j^k - \mathbf{p}^k\|} \tag{4}$$

Here $\beta_k > 1$ means the surounding normals converge slower to the target normal than the vertices do, meaning the surface is not G^2. On the other hand, $\beta_k < 1$ means the normals converge too quickly and we are likely generating a flat spot. So the ideal value is $\beta_k = 1$.

We used these numerical metrics to compare our LS^3 scheme for triangular meshes to the standard Loop scheme (*Std-Loop*), and two other variants proposed by Barthe et al. [BK04] exhibiting bounded curvature (*Curv-Loop*), and no polar artifact (*Polar-Loop*) respectively. For regular vertices (valence 6) and neglecting the very few first subdivision steps, in all cases we always found the ideal values $\rho_k = 0.5$, $\alpha_k = 0.5$, and $\beta_k = 1$ that is coherent with our 2D analysis of the Hölder regularity. In other words, our projection step seems to not compromise the C^2 continuity in the regular case.

Figure 6 shows worse case plots of these three metrics for thousands of random configurations generated around an

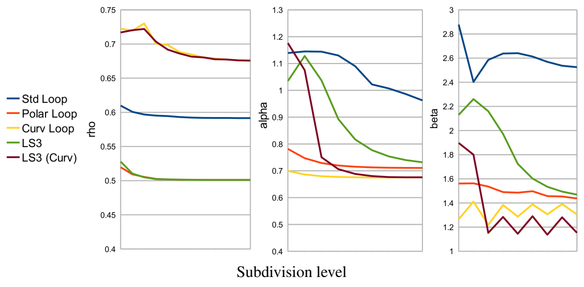

Figure 6: *From left to right, plots of our C^0, G^1, and G^2 numerical metrics for various schemes. They correspond to the worst case of 10k random configurations around an extraordinary vertex of valence 12.*

extraordinary vertex of valence 12. For comparison purpose, we also included a LS^3 variant using the bounded curvature masks [BK04] for both the relaxation and projection steps (noted LS^3-*Curv*). As can be seen, the LS^3 variants behave like their respective linear ones, meaning the projection step has a minimal impact on the continuity of the surface. Notice how polar artifacts are well detected by the C^0 metric ρ while the bounded curvature scheme clearly exhibits a better G^2 behavior. This is an important observation showing our numerical method matches well the theoretical results.

7. Implementation and Results

Implementation and performance

The complete LS^3 algorithm is summarized as pseudocode in figure 7 in the case of primal diadic subdivision of triangular meshes. This algorithm can be easily adapted to other refinement strategies. For instance, quadrangular meshes can be handled by adding an additional loop through the faces and using Catmull-Clark masks [CC78]. Figure 10 shows some results of Catmull-Clark and $\sqrt{3}$ [Kob00] subdivision enhanced with our LS^3 method. In summary, LS^3 subdivision mainly differs to standard subdivision by the additional projection step and the propagation of surface normals. It is therefore straightforward to adapt any existing implementation of Loop or Catmull-Clark subdivision to its respective LS^3 variant. Moreover, since the cost of the projection operator is in $O(n)$ (where n is the size of the mask), the theoretical complexity is of the same order as classical schemes. In practice, we observed that our implementation as a MeshLab [mes] plugin is half as fast than the initial implementation of the Loop scheme.

Initial normals

Our projection operator makes the final surface dependent on the given vertex normals. Vertex normals are usually computed as a weighted average of the adjacent face normals where classical choices for the weights include the face areas, the angles, or even Max's method [Max99]. While all these weighting strategies are based on very different interpretation of the underlying mesh connectivity, in practice the differences are seldom perceptible for sophisticated enough

```
for each vertex $\mathbf{p}_i^0$ of $M^0$ do
    compute its initial normal $\mathbf{n}_i^0$
for each subdivision step $k$ do
    for each vertex and edge $o$ do
        splitting rule:
            • create a new point $\mathbf{p}_i^k$ associated to $o$
        relaxation rule:
            • compute its intermediate position $\mathbf{q}_i^k$ by weighted
              averaging using the masks of fig. 3
        projection operator:
            • fit an algebraic sphere using eq. (1) and masks of fig. 3
            • set $\mathbf{p}_i^k$ as the projection of $\mathbf{q}_i^k$ onto the sphere
            • set its normal $\mathbf{n}_i^k$ as the gradient of the sphere at $\mathbf{p}_i^k$
    update the mesh topology yielding $M^k$
for each vertex $\mathbf{p}_i^n$ of $M^n$ do
    re-compute its final normal $\mathbf{n}_i^n$ using surrounding face normals
```

Figure 7: *Pseudocode for LS^3 subdivision.*

models. The results of this paper have been obtained using for the weights the face areas. The same method has been employed to compute the normals of the final refined meshes.

Note that Max's method [Max99] might be of some particular interest since the computed normals for a group of neighboring vertices lying on a sphere exactly match the sphere normals. This allows an exact reproduction of spherical parts.

Figure 8 shows interesting results can be achieved by allowing the user to edit the input normals. In practice this might be useful to fine tune the shape without increasing the mesh resolution. Interactive tools to edit these normals in an intuitive way have yet to be developed.

Fairness comparisons

In section 6 we found that LS^3 seems to possesses continuity properties comparable to weighted centroid schemes. However, this does not bring much insights about the fairness of the surface.

Figure 1 shows that LS^3 generates surfaces which are comparable to Loop's scheme everywhere the control mesh is regular but with a significantly improved behavior around extraordinary vertices. This is even more visible in the saddle example of Figure 9 where LS^3 is clearly superior to other variants of the Loop scheme and almost indistinguishable from the original quadric. Only a very small artifact is

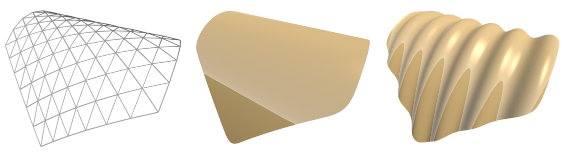

Figure 8: *A regular mesh (left) subdivided by LS^3 using default normal computation (middle), and after tweaking the normals (right).*

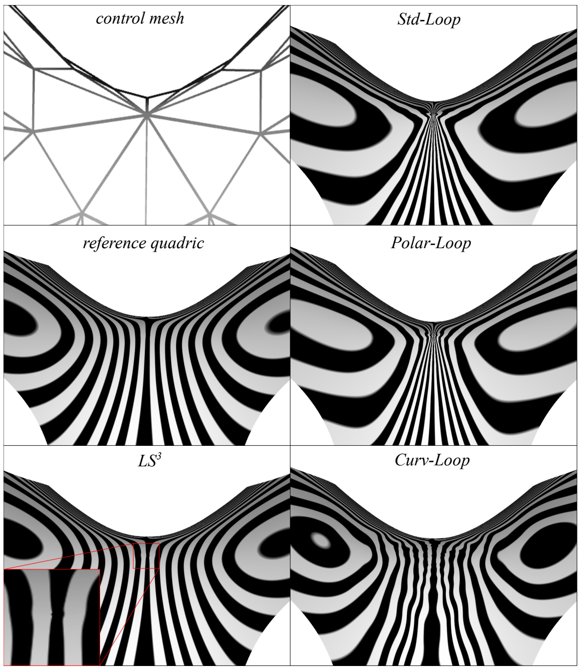

Figure 9: *Comparison of various Loop's variants on a saddle sampled around a vertex of valence* 12 *and using reflection lines to depict surface flaws.*

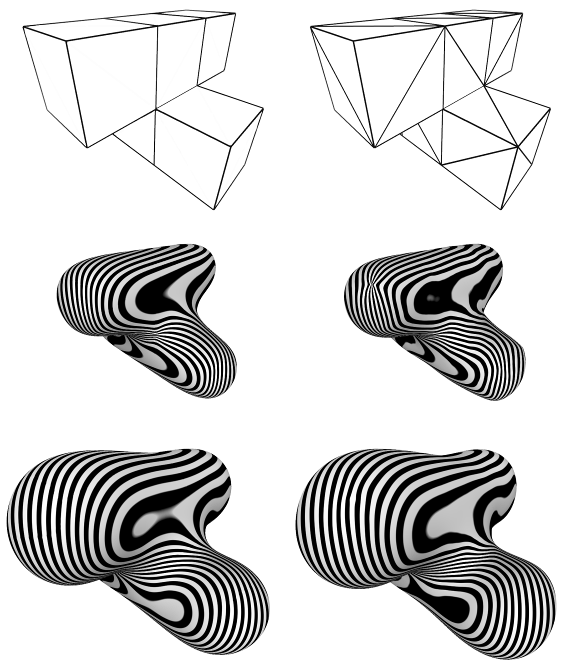

Figure 10: *Illustration of Catmull-Clark (left) and* $\sqrt{3}$ *subdivision (right) using the respective standard rules (middle row) and the* LS^3 *enhanced versions (bottom row). Note the shrinking effect of standard subdivisions.*

visible at the exact location of the extraordinary vertex. As expected by Karčiauskas et al. [KPR04], the bounded curvature variant exhibits severe fairness issues. Figure 10 shows that similar gain on the surface fairness is obtained when applying our LS^3 approach to Catmull-Clark and $\sqrt{3}$ schemes.

Figure 11 presents another classical test case where weighted centroid subdivisions exhibit much stronger ripples than LS^3. This example also clearly shows that LS^3 does not possess the convex hull property since the projection step can move vertices outside the convex hull of the mask. On the other hand, surfaces produced by LS^3 are usually closer to the control mesh making the interactive edition more intuitive. Another consequence is that details are better preserved. In practice, the more a region of the mesh is close to a sphere (or a plane), the more the limit surface will be close to interpolate the original mesh.

8. Discussions and Conclusion

We presented LS^3, a simple generalization of subdivision surfaces that considerably reduces surface fairness issues of weighted centroid schemes. LS^3 treats all vertices in a uniform manner and can handle boundaries and sharp edges. We believe that part of the success of our approach is due to an appropriate use vertex normals to locally approximate the surface by spheres and, consequently, smooth out the oscillations and ripples. The little disadvantage is that increases the influence region of a control point to a 3-ring neighborhood instead of only 2 rings for classical schemes. On the other hand, this brings new edition possibilities to the artists. Designing such a normal edition tool is an interesting and challenging area for future work.

Despite the simplicity of our approach, the resulting subdivision scheme is not linear anymore. As the main consequences, we currently lack a rigorous theoretical analysis, it is not possible to directly compute a point onto the limit surface [Sta98], and invariance to general affine transformations is not guaranteed. Note that LS^3 is invariant to translations, rotations, and uniform scalings. We also acknowledge that the sphere fitting can lead to a slight unpleasant growing effect when LS^3 tries to smooth a sharp and anisotropic region (e.g., as in figure 5-right). For all these reasons, LS^3 does not compete as a replacement of NURBS for CAD applications. On the other hand, in practice we found that these limitations are only visible in simple textbook cases, and they become seldom perceptible for soft and/or sophisticated enough models such as, e.g., characters. Therefore, we believe that for many applications, the outstanding ability of LS^3 to remove ripples and other subdivision artifacts, combined with its simplicity and efficiency largely compensate these limitations. This is especially true for game and movie industries where subdivision surfaces are already well established, thus enforcing the artists to spend a large amount of time to *hide* the extraordinary vertices of their models.

Our LS^3 approach also opens the door to many future researches, e.g. by investigating novel local approximation methods. For instance, moving from algebraic spheres to

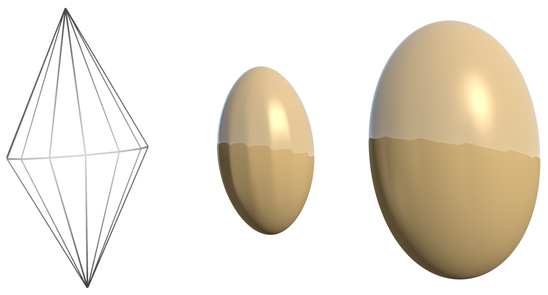

Figure 11: *Comparison of Loop's method (middle) and LS^3 (right) on a* 16*-sided double cone. Note the high shrinking effect and oscillations of the surface produced with Loop's scheme.*

generic quadric could potentially solve the aforementioned growing effect and bring affine invariance. However, our initial efforts in that direction showed that quadrics introduce more difficulties than solutions and good regularization terms have to be found.

Acknowledgment

The mannequin model of figure 1 is provided courtesy of MPII by the AIM@SHAPE Shape Repository. This work is part of the the SeARCH project, funded by the french National Research Agency ANR-09-CORD-019-01

References

[AA04] ALEXA M., ADAMSON A.: On normals and projection operators for surfaces defined by point sets. In *Proceedings of the Eurographics symposium on point-based graphics* (2004), pp. 149–156. 3

[AA06] ADAMSON A., ALEXA M.: Anisotropic point set surfaces. In *Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa* (2006), pp. 7–13. 4

[ABC*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics 9*, 1 (January 2003), 3–15. 2, 3

[AK04a] AMENTA N., KIL Y.: Defining point-set surfaces. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2004) 23*, 3 (2004), 264–670. 3

[AK04b] AMENTA N., KIL Y.: The domain of a point set surface. In *Proceedings of the Eurographics symposium on point-based graphics* (2004). 3

[BK04] BARTHE L., KOBBELT L.: Subdivision scheme tuning around extraordinary vertices. *Computer Aided Geometric Design 21*, 6 (July 2004), 561–583. 1, 3, 5

[CADS09] CASHMAN T. J., AUGSDÖRFER. U. H., DODGSON N. A., SABIN M. A.: NURBS with extraordinary points: high-degree, non-uniform, rational subdivision schemes. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2009) 28*, 3 (August 2009). 1, 2

[CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design 10*, 6 (November 1978), 350–355. 1, 6

[DKT98] DEROSE T., KASS M., TRUONG T.: Subdivision surfaces in character animation. In *Proceedings of ACM SIGGRAPH* (1998), pp. 85–94. 1, 5

[DLG90] DYN N., LEVINE D., GREGORY J. A.: A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics 9*, 2 (April 1990), 160–169. 1

[DS78] DOO D., SABIN M.: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design 10*, 6 (November 1978), 356–360. 1

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2007) 26*, 3 (July 2007), 23. 3

[GGG08] GUENNEBAUD G., GERMANN M., GROSS M.: Dynamic sampling and rendering of algebraic point set surfaces. *Computer Graphics Forum (Proceedings of Eurographics 2008) 27*, 2 (April 2008), 653–662. 2, 4

[Kob96] KOBBELT L.: Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum* (1996), vol. 15, pp. 409–420. 1

[Kob00] KOBBELT L.: $\sqrt{3}$-subdivision. In *Proceedings of ACM SIGGRAPH 2000* (2000). 1, 6

[KPR04] KARČIAUSKAS K., PETERS J., REIF U.: Shape characterization of subdivision surfaces: case studies. *Computer Aided Geometry Design 21*, 6 (July 2004), 601–614. 1, 6

[Kui98] KUIJT F.: *Convexity Preserving Interpolation - Stationary Nonlinear Subdivision and Splines*. PhD thesis, University of Twenty, 1998. 5

[Lev03] LEVIN D.: Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization* (2003), 37–49. 2

[Lev06] LEVIN A.: Modified subdivision surfaces with continuous curvature. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006) 25*, 3 (July 2006), 1035–1040. 2

[Loo87] LOOP C.: *Smooth subdivision surfaces based on triangles*. Master's thesis, University of Utah, 1987. 1

[Max99] MAX N.: Weights for computing vertex normals from facet normals. *Journal of Graphics Tools 4*, 2 (March 1999), 1–6. 6

[MDL05] MARINOV M., DYN N., LEVIN D.: Geometrically controlled 4-point interpolatory schemes. *Advances in Multiresolution for Geometric Modelling* (2005), 301–315. 5

[mes] Meshlab. Visual Computing Lab, ISTI, CNR, url: `http://meshlab.sourceforge.net/`. 6

[PR97] PETERS J., REIF U.: The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics 16*, 4 (October 1997), 420431. 1

[Rei96] REIF U.: A degree estimate for polynomial subdivision surfaces of higher regularity. *Proceedings of American Mathematical Society 124*, 7 (1996), 2167. 1

[SB02] SABIN M., BARTHE L.: Artifacts in recursive subdivision surface. In *Proceedings of Curves and Surfaces* (2002), pp. 353–362. 1

[Sta98] STAM J.: Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of ACM SIGGRAPH* (1998), ACM, pp. 395–404. 7

[WW02] WARREN J., WEIMER H.: *Subdivision methods for geometric design: a constructive approach*. Morgan Kaufmann, 2002. 1

[Zor06] ZORIN D.: Modeling with multiresolution subdivision surfaces. In *Proc. ACM SIGGRAPH [Courses]* (2006). 1

[ZS00] ZORIN D., SCHRÖEDER P.: Subdivision for modeling and animation. In *Proc. ACM SIGGRAPH [Courses]* (2000). 1, 5

[ZSS97] ZORIN D., SCHRÖDER P., SWELDENS W.: Interactive multiresolution mesh editing. In *Proc. ACM SIGGRAPH* (1997). 1