RIAS: Repeated Invertible Averaging for Surface Multiresolution of Arbitrary Degree

Troy Alderson⬤, Ali Mahdavi-Amiri, and Faramarz Samavati

Abstract—In this article, we introduce two local surface averaging operators with local inverses and use them to devise a method for surface multiresolution (subdivision and reverse subdivision) of arbitrary degree. Similar to previous works by Stam, Zorin, and Schröder that achieved forward subdivision only, our averaging operators involve only direct neighbours of a vertex, and can be configured to generalize B-Spline multiresolution to arbitrary topology surfaces. Our subdivision surfaces are hence able to exhibit C^d continuity at regular vertices (for arbitrary values of d) and appear to exhibit C^1 continuity at extraordinary vertices. Smooth reverse and non-uniform subdivisions are additionally supported.

Index Terms—Geometric modeling, subdivision, multiresolution

✦

1 INTRODUCTION

IT is said that "Simplicity is the ultimate sophistication," yet the principle of simplicity often offers benefits aside from any philosophical beauty. Within the realm of mesh editing, for instance, the simplicity of editing operations (or lack thereof) has impacts on understandability, ease of implementation, and runtime efficiency. However, it is not always clear how complex goals can be achieved through simple means.

Generating smooth curves and meshes is one such goal that at first glance would not appear to have a simple solution. Yet the Lane-Riesenfeld algorithm [1] and its generalization to surfaces by Stam [2] and Zorin and Schröder [3] offer a comprehensive solution to smooth subdivision based on simple operations. Unlike typical subdivision algorithms that converge to B-Spline curves/surfaces at the limit, this "classical approach" allows one to select the degree of the resulting surface (i.e., the degree of the associated B-Spline basis functions). Hence, a curve/surface with a finite but controllable level of continuity (i.e., arbitrary smoothness) can be generated without implementing alternative algorithms.

For surfaces in particular, the classical approach enjoys two particularly desirable properties:

- *Simplicity/Locality*: Much like the Lane-Riesenfeld algorithm, the method is composed of simple operations: an initial topology subdivision followed by several averaging steps. These operations involve only the local direct neighbours (i.e., the one-ring) of each vertex, and as a result, are not only efficient but also easy to implement.

- *C^d Continuity at Regular Vertices*: At regular vertices, the subdivision rules reduce to tensor product B-Spline subdivision rules of arbitrary bi-degree. Hence, the method is arbitrarily smooth at regular vertices.

However, unlike the widely-used Catmull-Clark [4] and Doo-Sabin [5] schemes, the classical approach cannot be reversed. This is because its averaging steps are not invertible, and so cannot be undone in a similarly general way. As a result, reverse subdivision schemes must be implemented on a case-by-case basis if one wishes to support applications such as level-of-detail rendering, compression, or multiscale editing.

In this paper, we propose a similarly comprehensive approach to surface subdivision, reverse subdivision, and multiresolution. Our subdivision can produce C^d *continuity at regular vertices*, but can also be *reversed* in a general manner using *simple operations* that involve only direct neighbours of a vertex. The high-resolution version of a reverse subdivided mesh can additionally be *reconstructed* on-demand.

Our work extends the methodology of [6], which operates on spherical curves. In their method, which we refer to as *repeated invertible averaging (RIA)*, Alderson *et al.* replaced the Lane-Riesenfeld algorithm's averaging operator with invertible variants whose inverses involve only direct neighbours of a vertex. Each averaging operator moves a vertex towards an average of its neighbours, controlled using an associated weight value. Their method was shown to be able to reproduce B-Spline curve multiresolution up to degree 3.

We present a generalized framework for *repeated invertible averaging on surfaces (RIAS)*, which we use to construct multiresolution representations for general topology surfaces. Novel contributions of RIAS include:

- Two surface averaging operators defined on direct vertex neighbours with inverses also defined on direct vertex neighbours. The properties of these operators can be adjusted using weighting parameters.
- A class of surface subdivision and reverse subdivision schemes that includes B-Spline schemes of *any* degree.

- *T. Alderson and F. Samavati were with the University of Calgary in Calgary, Calgary, AB T2N 1N4, Canada.*
  E-mail: {tfalders, samavati}@ucalgary.ca.
- *A. Mahdavi-Amiri was with the Simon Fraser University in Burnaby, Burnaby, BC V5A 1S6, Canada. E-mail: amahdavi@sfu.ca.*

TABLE 1
Summary of RIAS's Features Compared to Prior Works

| | Arbitrary Degree | Smooth Reverse | | Non-Uniform | Compact MR |
		Primal	Dual		
RIAS	Yes	Yes	Yes	Yes	Yes
[2], [3]	Yes	N/A	N/A	No	N/A
[9], [10]	No	No	No	No	Yes
[7]	No	Yes	No	No	No
[8]	No	Yes	No	No	Yes

- A recipe for the weighting parameters that configures our method to reproduce these B-Spline schemes. These weighting parameters can be blended in order to blend between different schemes.
- Constructions for smooth reverse, non-uniform, and adaptive subdivision that can be reversed on demand.

RIAS supports several existing concepts from the literature (refer to Table 1), offering a single framework that can be used to define arbitrary degree subdivision, smooth reverse subdivision (see [7], [8]), as well as non-uniform and adaptive subdivision—all of which can be reversed and restored on demand. Its generality and comprehensiveness allows us to craft a variety of schemes by tailoring the weighting parameters to target specific requirements, and can be used to underpin practical applications such as multiscale editing in CAD (where smooth surfaces of up to C^7 continuity are regularly used [2]) or mesh compression. The framework additionally offers significant speed gains over the classical approach of [2], [3], as vertex positions do not need to be cached and our operations can be parallelized on the GPU with minimal overhead.

After discussing some related work in Section 2, we describe our invertible averaging operators and surface multiresolution schemes in Section 3. This is followed in Section 4 by discussions on how RIAS can be used to support smooth reverse and non-uniform subdivision. Considerations on the selection of weighting parameters are presented in Section 5. We then remark on the continuity of our subdivision surfaces at extraordinary vertices in Section 6. Finally, we present our results and conclude the paper in Sections 7 and 8.

2 RELATED WORK

The earliest subdivision schemes for general topology surfaces were proposed in 1978 by Doo and Sabin [5] and Catmull and Clark [4]. Others, including Loop subdivision [11] and Simplest subdivision [12], soon followed. Since then, and despite advances in the field (see [13] for a survey), Catmull-Clark subdivision has become a go-to standard for surface subdivision due to its simplicity and C^2 continuity at regular vertices.

While the tensor product connectivity of regular vertices allows for simple continuity analysis, the analysis at extraordinary vertices has always been much more difficult. In [14], Reif introduced the characteristic map, which can be used to rigorously analyze the continuity of surface subdivision schemes at extraordinary vertices. Given a local subdivision matrix S for a surface subdivision scheme, the characteristic map ψ parametrizes the limit subdivision surface using the subdominant eigenvectors of S.

From Reif's work, necessary and sufficient conditions for continuity at extraordinary vertices — largely dependent on

the eigenvalues of S — were derived (see [15] and [16, Chapter 8]). In [17], it was shown that Catmull-Clark subdivision and Doo-Sabin subdivision are both C^1 at extraordinary vertices.

Though subdivision is an important tool within the field of geometric modeling, the usefulness of multiresolution frameworks and the ability to reverse subdivision has also been recognized [7], [18]. Reverse schemes have been proposed for Catmull-Clark [10], [19], Doo-Sabin [20], and Loop subdivision [9]. As reverse subdivision schemes are not unique and developing multiresolution frameworks with nice properties is difficult, work is ongoing to improve the behaviour of different schemes and the appearance of the resulting surfaces [8], [21].

In 2001, Stam [2] and Zorin and Schröder [3] introduced a subdivision algorithm designed to exhibit arbitrary continuity at regular vertices, though the continuity at extraordinary vertices remains C^1. Their construction is based on a generalization to surfaces of the Lane-Riesenfeld algorithm [1], which reproduces B-Spline curve subdivision of any arbitrary degree. Regrettably, neither algorithm has a known reverse scheme. This becomes problematic when we wish to represent the curves or surfaces resulting from these algorithms at multiple levels of detail.

While curve subdivision schemes can be reversed using optimization on a given subdivision matrix [18], [22], this method does not generalize well to surface subdivision schemes or non-euclidean spaces. For this reason, Alderson et al. [6], [23] explored modifying the Lane-Riesenfeld algorithm using local smoothing operators with local inverses (see [8]) in order to establish multiresolution for curves on the surface of a sphere in a general and efficient way. Our work generalizes this repeated invertible averaging method to surface subdivision, much as Stam, Zorin, and Schröder generalized the Lane-Riesenfeld algorithm.

3 RIAS FOR MULTIRESOLUTION SURFACES

In this section, we present our invertible averaging operators for surfaces and the multiresolution schemes constructed using them.

At a high level, these schemes work as follows. To increase the resolution of a mesh, we start with an initial subdivision/refinement (e.g., linear subdivision) and then apply several (say, m) iterations of our averaging operators (see Figs. 1 and 3). The reverse subdivision consists of applying m iterations of the inverse averaging operators followed by a basic reverse subdivision scheme corresponding to the initial subdivision.

The number of averaging operations applied, m, is linked to the support of the scheme (and its bi-degree, when the subdivision reproduces polynomial surfaces at the limit — as in the case of B-Spline subdivision). Similarly to the works of [2] and [6], we divide our schemes into odd degree (primal) schemes and even degree (dual) schemes.

Before delving into the specifics of these schemes, we first introduce some basic terminology and remark on the design of averaging operators with local inverses.

3.1 Terminology

A mesh $M = (V, E, F)$ consists of a set of vertices V, a set of edges E, and a set of faces F. We denote by $v \in V$ a vertex in 3D space, $\boldsymbol{e} = (v_0, v_1) \in E$ an edge connecting vertices v_0
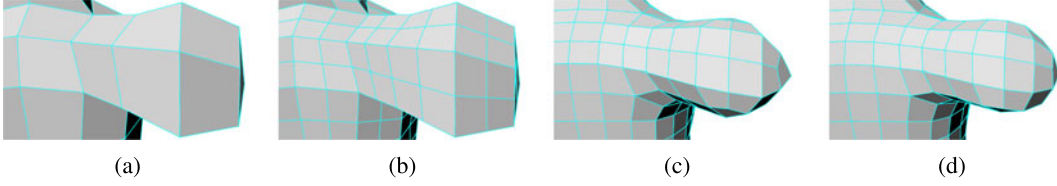
Fig. 1. Primal subdivision applied to a mesh. (a) A hand of the coarse Teddy mesh (see Fig. 8). (b) An initial face split is applied to the mesh. (c) Λ_0 moves the vertex- and edge-vertices while fixing the face-vertices. (d) Λ_1 moves the face- and edge-vertices while fixing the vertex-vertices.

and v_1, and $\boldsymbol{f} = (v_0, v_1, \ldots, v_{n-1})$ a face with $n \geq 3$ vertices $v_0, v_1, \ldots, v_{n-1}$.

A vertex q is said to be connected to a vertex p if there exists an edge $e \in E$ such that $p, q \in \boldsymbol{e}$. Vertex q is said to be a direct neighbour of p (or in the one-ring of p) if there exists a face $\boldsymbol{f} \in F$ such that $p, q \in \boldsymbol{f}$. The neighbourhood $N(v)$ of vertex v is the set of all direct neighbours of v. The valence of a vertex v is the number of edges $e \in E$ such that $v \in \boldsymbol{e}$.

A subdivision scheme P is a mapping from the mesh M to a new mesh $M' = (V', E', F')$ with $|V'| > |V|$. A reverse subdivision scheme $\hat{P}$ maps a mesh M to a new mesh $M' = (V', E', F')$ with $|V'| < |V|$. The reverse scheme $\hat{P}$ is considered to reverse a subdivision scheme P if $\hat{P} \circ P(M) = M$ for any mesh M.

3.2 Remarks on Local Inverses

Our averaging operators, inspired by those of [6], have the property that both the operator and its inverse are both *local* (i.e., operate on direct neighbours of a vertex only), in contrast to other averaging operators that do not guarantee locality of the inverse (if, indeed, they are invertible at all).

The core obstacle to ensuring locality of the inverse lies in the *co-dependency* of vertex positions after averaging. Consider two neighbouring vertices p and q that are moved to new positions p' and q' by an averaging operator, with

$$p' = a_1 p + a_2 q + \cdots, \qquad q' = b_1 p + b_2 q + \cdots.$$

Formula rearrangement gives the original positions p and q as

$$p = \frac{1}{a_1}p' - \frac{a_2}{a_1}q - \cdots, \qquad q = \frac{1}{b_2}q' - \frac{b_1}{b_2}p - \cdots.$$

These formulae are co-dependent: finding p depends on q, but finding q also depends on p. Therefore, q' and p' in the formulae for p and q must be replaced by different combinations of points, destroying the locality of the original formulae. Indeed, returning the original p and q can depend on every vertex position in the averaged mesh, or, if the operator is not invertible, become impossible.

Our averaging steps avoid such co-dependency: if finding an original vertex position p depends on q, then q can be found without knowledge of p. Inevitably, this requires knowledge of a set of "fixed points" — point positions that are unchanged by the averaging. (Note that these need not be explicit vertices in the mesh.) Smoothing of the entire mesh is accomplished by varying the set of fixed points with every application of the averaging operators.

3.3 Primal Schemes

Primal schemes like Catmull-Clark subdivision [4] map vertices in the coarse mesh to vertices in the subdivided mesh (called "vertex-vertices"). Edges and faces are split by the introduction of new edge- and face-vertices.

The initial subdivision employed for our primal schemes, denoted by I_P, generalizes linear subdivision and is identical to the face split employed by Catmull & Clark (see Fig. 1b). Edge-vertices are introduced at the midpoint of each mesh edge, and face-vertices are introduced at the centroid of each mesh face. Formally

$$VV = \underbrace{\{v \mid v \in V\}}_{\text{vertex-vertices}},$$

$$EV = \underbrace{\{\frac{1}{2}v_0 + \frac{1}{2}v_1 \mid \boldsymbol{e} = (v_0, v_1) \in E\}}_{\text{edge-vertices}},$$

$$FV = \underbrace{\{\frac{1}{n}\sum_{i=0}^{n-1} v_i \mid \boldsymbol{f} = (v_0, v_1, \ldots, v_{n-1}) \in F\}}_{\text{face-vertices}},$$

$$I_P(V) = VV \cup EV \cup FV.$$

This initial subdivision[1] is followed by $m \geq 0$ averaging steps Λ_k ($0 \leq k < m$) that map old vertex positions to new vertex positions (see Figs. 1c and 1d). In order to guarantee an inverse that operates on direct neighbours, Λ_k fixes an alternating set of vertices in each iteration.

Each averaging step Λ_k is controlled by a smoothing weight $0 < s_k < 1$ that determines how strongly a vertex is moved to a combination of its neighbours. We denote the set of weights as $S = \{s_0, s_1, \ldots, s_{m-1}\}$. Given a set of weights S, our primal subdivision scheme, P_{Primal}, is given by

$$P_{Primal}(V, E, F) = \Lambda_{m-1} \circ \cdots \circ \Lambda_1 \circ \Lambda_0 \circ I_P(V, E, F).$$

To define the action of Λ_k, we will use the superscript $[k]$ to indicate vertex positions or sets of vertex positions after k averaging steps have been applied.[2] For example

$$v^{[0]} \in V^{[0]} = I_P(V),$$
$$v^{[k]} \in V^{[k]} = \Lambda_{k-1} \circ \cdots \circ \Lambda_1 \circ \Lambda_0 \circ I_P(V).$$

Given a vertex $v^{[k]}$, we will also let $v_{avg}^{[k']}(v^{[k]})$, $e_{avg}^{[k']}(v^{[k]})$, and $f_{avg}^{[k']}(v^{[k]})$ respectively denote the average of the vertex-vertices, edge-vertices, and face-vertices in the one-ring around $v^{[k]}$. That is

---

1. As with Catmull-Clark, the faces of the resulting mesh are all quads and the edge-vertices are all valence 4. The reverse can be applied to quad meshes whose vertices can be partitioned into VV, EV, and FV.

2. It is important to note that, unlike many works on subdivision, the $[k]$ superscript does not represent the number of subdivisions applied. An application of subdivision is completed when $[k] = [m]$.
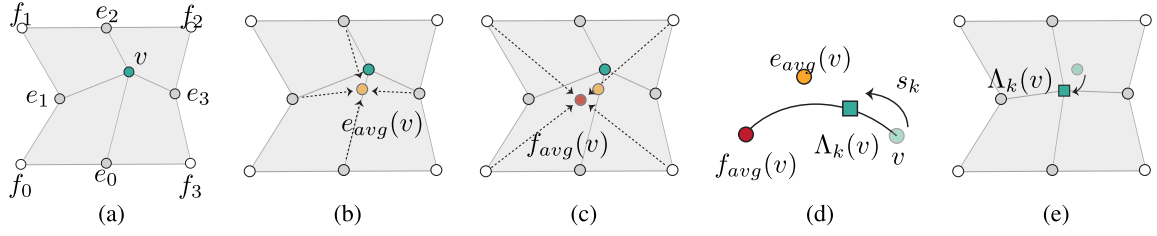
Fig. 2. Illustration of Λ_k (with even k) acting on a vertex-vertex. (a) A vertex-vertex v with edge-vertices (labelled e_0 through e_3) and face-vertices (labelled f_0 through f_3) in its one-ring. (b) The average of the edge-vertices, $e_{avg}(v)$, is found. (c) The average of the face-vertices, $f_{avg}(v)$, is found. (d) The new position of v, $\Lambda_k(v)$, lies on a quadratic Bezier curve between v, $e_{avg}(v)$, and $f_{avg}(v)$. (e) The vertex-vertex v is moved to $\Lambda_k(v)$.

$$v_{avg}^{[k']}(v^{[k]}) = < \left\{ p^{[k']} \mid p^{[k]} \in VV^{[k]} \cap N(v^{[k]}) \right\} >,$$
$$e_{avg}^{[k']}(v^{[k]}) = < \left\{ p^{[k']} \mid p^{[k]} \in EV^{[k]} \cap N(v^{[k]}) \right\} >,$$
$$f_{avg}^{[k']}(v^{[k]}) = < \left\{ p^{[k']} \mid p^{[k]} \in FV^{[k]} \cap N(v^{[k]}) \right\} >,$$

where $< \cdot >$ denotes the average of a set. (The distinction between $[k]$ and $[k']$ is relevant to the definition of Λ_k^{-1}, where we must refer to the original vertex positions.)

Now, when we apply Λ_k and k is even, we fix the face-vertex positions $f^{[k+1]} \in \Lambda_k(FV^{[k]})$ and compute new edge-vertex positions $e^{[k+1]} \in \Lambda_k(EV^{[k]})$ and new vertex-vertex positions $v^{[k+1]} \in \Lambda_k(VV^{[k]})$. These are given by

$$
\begin{aligned}
f^{[k+1]} &= f^{[k]}, \\
e^{[k+1]} &= (1 - s_k) \cdot e^{[k]} + s_k \cdot f_{avg}^{[k]}(e^{[k]}),
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
v^{[k+1]} = (1 - s_k)^2 \cdot v^{[k]} + 2s_k(1 - s_k) \cdot e_{avg}^{[k]}(v^{[k]}) \\
+ s_k^2 \cdot f_{avg}^{[k]}(v^{[k]}).
\end{aligned}
\tag{2}
$$

See Fig. 2 for an illustration. When k is odd, we instead fix the vertex-vertex positions and compute new edge- and face-vertex positions. These are given by

$$
\begin{aligned}
v^{[k+1]} &= v^{[k]}, \\
e^{[k+1]} &= (1 - s_k) \cdot e^{[k]} + s_k \cdot v_{avg}^{[k]}(e^{[k]}), \\
f^{[k+1]} &= (1 - s_k)^2 \cdot f^{[k]} + 2s_k(1 - s_k) \cdot e_{avg}^{[k]}(f^{[k]}) \\
&\quad + s_k^2 \cdot v_{avg}^{[k]}(f^{[k]}).
\end{aligned}
$$

Note that the edge-vertices are moved in each iteration, whereas the vertex- and face- vertices are moved in only half, and that only two vertex- or face-vertices are ever in the one-ring around an edge-vertex. In order to avoid caching the original edge-vertex positions $e^{[k]}$, the $e^{[k+1]}$ are calculated after the $v^{[k+1]}$ and $f^{[k+1]}$.

The inverse of Λ_k follows from formula rearrangement. If k is even, then Λ_k^{-1} fixes the face-vertices $f^{[k]} \in \Lambda_k^{-1}(FV^{[k+1]})$ and computes the original edge-vertex positions $e^{[k]} \in \Lambda_k^{-1}(EV^{[k+1]})$ and original vertex-vertex positions $v^{[k]} \in \Lambda_k^{-1}(VV^{[k+1]})$

$$
\begin{aligned}
f^{[k]} &= f^{[k+1]}, \\
e^{[k]} &= \frac{1}{1 - s_k} \cdot e^{[k+1]} - \frac{s_k}{1 - s_k} \cdot f_{avg}^{[k]}(e^{[k+1]}),
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
v^{[k]} = \left(\frac{1}{1 - s_k} \right)^2 \cdot v^{[k+1]} - \frac{2s_k}{1 - s_k} \cdot e_{avg}^{[k]}(v^{[k+1]}) \\
- \left(\frac{s_k}{1 - s_k} \right)^2 \cdot f_{avg}^{[k]}(v^{[k+1]}).
\end{aligned}
\tag{4}
$$

If k is odd, we have

$$
\begin{aligned}
v^{[k]} &= v^{[k+1]}, \\
e^{[k]} &= \frac{1}{1 - s_k} \cdot e^{[k+1]} - \frac{s_k}{1 - s_k} \cdot v_{avg}^{[k]}(e^{[k+1]}), \\
f^{[k]} &= \left(\frac{1}{1 - s_k} \right)^2 \cdot f^{[k+1]} - \frac{2s_k}{1 - s_k} \cdot e_{avg}^{[k]}(f^{[k+1]}) \\
&\quad - \left(\frac{s_k}{1 - s_k} \right)^2 \cdot v_{avg}^{[k]}(f^{[k+1]}).
\end{aligned}
$$

When applying Λ_k^{-1}, we calculate the $e^{[k]}$ before the $v^{[k]}$ and $f^{[k]}$.

Given a reverse subdivision operator $\hat{I}_P$ that reverses the initial subdivision I_P, our primal subdivision scheme P_{Primal} can now be reversed using a reverse primal scheme $\hat{P}_{Primal}$

$$\hat{P}_{Primal}(V, E, F) = \hat{I}_P \circ \Lambda_0^{-1} \circ \Lambda_1^{-1} \circ \cdots \circ \Lambda_{m-1}^{-1}(V, E, F).$$

While $\hat{I}_P$ is not unique, one option is to combine the faces that were split by deleting the edge- and face-vertices. In order to reconstruct the mesh, we can map these vertices to multiresolution detail vectors. In this case, edge-vertices $e^{[0]}$ are mapped to details $\vec{d}_e$

$$e^{[0]} \rightarrow \vec{d}_e = e^{[0]} - v_{avg}^{[0]}(e^{[0]}),$$

and face-vertices $f^{[0]}$ are mapped to details $\vec{d}_f$

$$f^{[0]} \rightarrow \vec{d}_f = f^{[0]} - v_{avg}^{[0]}(f^{[0]}).$$

Details are then restored during subdivision by adding them to the edge- and face-vertex positions immediately following the initial subdivision I_P.

An alternative choice for $\hat{I}_P$ that produces coarse meshes with fewer artifacts can be found by taking inspiration from the local least squares method of [22] (as applied to linear curve subdivision). The resulting reverse subdivision operation can be interpreted as translating the fine vertex positions by scaled detail vectors to produce the new coarse vertex positions. Similarly, we can map edge-vertices $e^{[0]}$ and face-vertices $f^{[0]}$ to details d_e and d_f as above, with the coarse vertices v' calculated by translating each vertex-vertex $v^{[0]}$ by scaled versions of the details in its one-ring neighbourhood
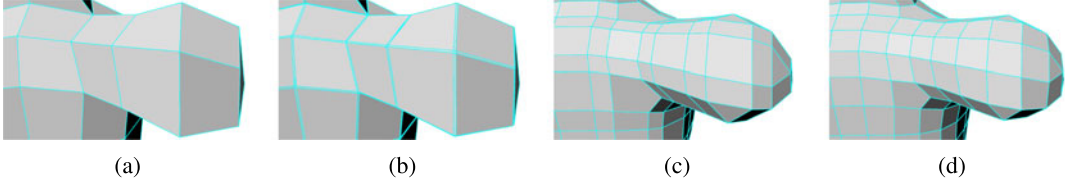
Fig. 3. Dual subdivision applied to a mesh. (a) A hand of the coarse Teddy mesh (see Fig. 8). (b) An initial vertex split is applied to the mesh. The face-faces are slightly shrunken for illustrative purposes. (c) Φ_0 shrinks the face-faces, retaining their centroids. (d) Φ_1 shrinks the vertex-faces, retaining their centroids.
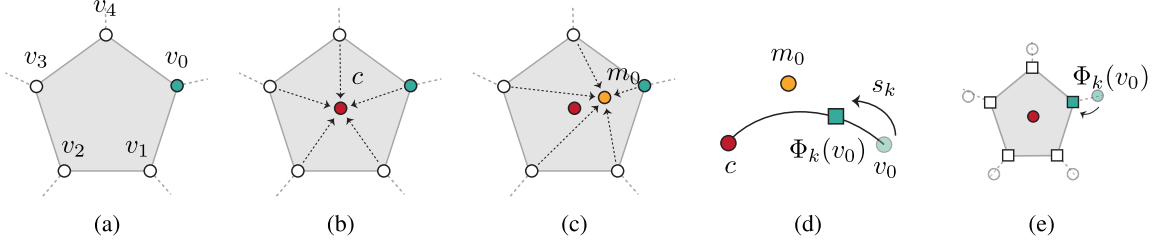

Fig. 4. Illustration of Φ_k acting on a face. (a) A shrinking face with five vertices (labelled v_0 through v_4). (b) The centroid c of the face is found. (c) An intermediate point m_i is found for every vertex v_i. (d) The new position of each v_i, $\Phi_k(v_i)$, lies on a quadratic Bezier curve between v_i, m_i, and c. (e) The vertices v_i are each moved to $\Phi_k(v_i)$, shrinking the face.

$$v^{[0]} \rightarrow v' = v^{[0]} + \frac{1}{3}\sum \vec{d}_e + \frac{1}{9}\sum \vec{d}_f.$$

During reconstruction, these scaled detail vectors are subtracted from the coarse positions v' before the initial subdivision I_P is applied.

Pseudocode for our primal multiresolution schemes can be found in supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2020.2972877.

3.4 Dual Schemes

Dual schemes like Doo-Sabin subdivision [5] map faces in the coarse mesh to faces in the subdivided mesh (called "faces-faces"). Edges and vertices are replaced with new edge- and vertex-faces.

The initial subdivision I_D employed for our dual schemes is a vertex split method that generalizes Haar subdivision [24] to surfaces. The connectivity is identical to a mesh subdivided with Doo-Sabin subdivision, but the vertex positions are unchanged from vertices in the coarse mesh (see Fig. 3b). Formally

$$VF = \underbrace{\{\overbrace{(v, v, \ldots, v)}^{n \text{ copies}} \mid v \in V \text{ of valence } n\}}_{\text{vertex-faces}},$$

$$EF = \underbrace{\{(v_0, v_0, v_1, v_1) \mid e = (v_0, v_1) \in E\}}_{\text{edge-faces}},$$

$$FF = \underbrace{\{\boldsymbol{f} \mid \boldsymbol{f} = (v_0, v_1, \ldots, v_{n-1}) \in F\}}_{\text{face-faces}},$$

$$I_D(F) = VF \cup EF \cup FF.$$

This initial subdivision[3] is followed by $m \geq 0$ face-shrinking operations Φ_k ($0 \leq k < m$) that map old vertex positions to new vertex positions (see Figs. 3c and 3d). In

order to ensure the inverse is local, we shrink an alternating set of faces in each iteration, such that each face and its shrunken version share the same centroid (i.e., the centroids are fixed points). Our inverse operators Φ_k^{-1} can hence expand faces out from these fixed centroids, similarly to the method used in [20] to reverse Doo-Sabin subdivision.

As in our primal schemes, each Φ_k is controlled by a smoothing weight in $S = \{s_0, s_1, \ldots, s_{m-1}\}$. Given these weights S, our dual subdivision scheme, P_{Dual}, is given by

$$P_{Dual}(V, E, F) = \Phi_{m-1} \circ \cdots \circ \Phi_1 \circ \Phi_0 \circ I_D(V, E, F).$$

Again we use the superscript $[k]$ to indicate vertex positions after k averaging steps have been applied.

Each Φ_k is defined as follows. If k is even, then we shrink the face-faces $\boldsymbol{f}^{[k]} \in FF^{[k]}$; otherwise, we shrink the vertex-faces $\boldsymbol{f}^{[k]} \in VF^{[k]}$. For each shrinking face $\boldsymbol{f}^{[k]} = (v_0^{[k]}, v_1^{[k]}, \ldots, v_{n-1}^{[k]})$, we find the centroid of the face

$$c^{[k]} = \frac{1}{n}\sum_{i=0}^{n-1} v_i^{[k]}, \tag{5}$$

and an intermediate point m_i for each vertex $v_i^{[k]}$

$$m_i^{[k]} = \sum_{j=0}^{n-1} \alpha_{i,j} \cdot v_j^{[k]}, \tag{6}$$

where the $\alpha_{i,j}$ are coefficients inspired by Doo-Sabin subdivision

$$\alpha_{i,j} = \frac{1 + \cos(2\pi(j-i)/n)}{n}.$$

Each vertex is then moved to a new position $v_i^{[k+1]} \in \Phi_k(V^{[k]})$ (see Fig. 4) given by

$$v_i^{[k+1]} = (1 - s_k)^2 \cdot v_i^{[k]} + 2s_k(1 - s_k) \cdot m_i^{[k]} + s_k^2 \cdot c^{[k]}. \tag{7}$$

The inverse of Φ_k is a face expansion operator that expands the face-faces $\boldsymbol{f}^{[k+1]} \in FF^{[k+1]}$ when k is even;

---

3. As with Doo-Sabin, the vertices of the resulting mesh are all valence 4 and the edge-faces are all quads. The reverse can be applied to meshes whose faces can be partitioned into VF, EF, and FF.

otherwise it expands the vertex-faces $f^{[k+1]} \in VF^{[k+1]}$. In order to apply Φ_k^{-1}, we must first determine the original set of intermediate points $m_i^{[k]}$ from the new vertex positions $v_i^{[k+1]}$ (the centroid is the same for both the original and the shrunken face, i.e., $c^{[k]} = c^{[k+1]}$)

$$m_i^{[k]} = \frac{1}{1 - s_k} \cdot \sum_{j=0}^{n-1} \alpha_{i,j} \cdot v_j^{[k+1]} - \frac{s_k}{1 - s_k} \cdot c^{[k]}. \tag{8}$$

Afterwards, the vertices $v_i^{[k+1]}$ may be returned to their coarse positions

$$v_i^{[k]} = \left(\frac{1}{1 - s_k}\right)^2 \cdot v_i^{[k+1]} - \frac{2s_k}{1 - s_k} \cdot m_i^{[k]} - \left(\frac{s_k}{1 - s_k}\right)^2 \cdot c^{[k]}. \tag{9}$$

Now, given a reverse scheme $\hat{I}_D$ that reverses I_D, we can define a reverse subdivision scheme $\hat{P}_{Dual}$ that reverses P_{Dual}

$$\hat{P}_{Dual}(V, E, F) = \hat{I}_D \circ \Phi_0^{-1} \circ \Phi_1^{-1} \circ \cdots \circ \Phi_{m-1}^{-1}(V, E, F).$$

Since I_D creates new faces by splitting vertices and edges, $\hat{I}_D$ should eliminate the vertex- and edge- faces by collapsing them down to vertices and edges. This involves deleting each vertex-face $f^{[0]} = (v_0^{[0]}, v_1^{[0]}, \ldots v_{n-1}^{[0]}) \in VF^{[0]}$ and all but one of its vertices, say $v_0^{[0]}$. This vertex is placed at a collapsed position v_0', for instance the face's centroid, i.e.,

$$v_0^{[0]} \rightarrow v_0' = \frac{1}{n} \sum_{i=0}^{n-1} v_i^{[0]},$$

while the other $n-1$ vertices ($v_i^{[0]}$ for $1 \le i < n$) are mapped to detail vectors

$$v_i^{[0]} \rightarrow \vec{d}_{v_i} = v_i^{[0]} - v_0'.$$

Any face-faces that were incident to $f^{[0]}$ become incident to v_0', and any edge-faces that were incident to $f^{[0]}$ become edges incident to v_0'. In order to reconstruct the mesh, during I_D every vertex v (of valence n) is split into a new vertex face

$$\boldsymbol{v} = \left(v - \sum_{i=1}^{n-1} \vec{d}_{v_i}, v + \vec{d}_{v_1}, \ldots, v + \vec{d}_{v_{n-1}}\right) \in VF^{[0]}.$$

Pseudocode for our dual multiresolution schemes (as well as a proof of the correctness of Equation (8)) can be found in supplementary material, available online.

4 RIAS VARIANTS

In addition to the main RIAS framework, we additionally propose two easy-to-implement modifications that can be used to achieve more complex behaviours. The first of these incorporates smooth reverse subdivision into our multiresolution process by using the averaging operators Λ_k and Φ_k to smooth reverse subdivided meshes. The second defines non-uniform subdivision and reverse subdivision processes by using different smoothing weights per vertex.

4.1 Smooth Reverse Subdivision

The reverse subdivision schemes $\hat{P}_{Primal}$ and $\hat{P}_{Dual}$ defined above will return the original coarse mesh when applied to a subdivided mesh. However, if the mesh is not the result of a subdivision process or if it has been modified from a subdivided mesh, then shape exaggerations will result (and are particularly significant when using higher degree schemes).

Different solutions to this problem exist. For instance, it is possible to customize $\hat{I}_P$ or $\hat{I}_D$ in order to obtain more desirable behaviour, as in Section 3.3. One may also scale down the weights (e.g., by a factor of $\frac{1}{2}$) in order to reduce the impact of each inverse averaging step. A third approach is to consider *smooth reverse subdivision* schemes that aim to reduce exaggerations in the coarse shape. As noted in [7], [8], smooth reverse frameworks apply smoothing operations after each application of a given reverse subdivision operation. These smoothing steps are then undone prior to reconstructing the original mesh.

Our averaging operators Λ_k and Φ_k, by virtue of their invertibility, are a natural choice for use in a smooth reverse framework, and they possess properties that address limitations with the schemes of [7], [8]. First, Λ_k and Φ_k are perfectly invertible and can be undone without additional information, unlike the method of [7]. Second, while they require subdivision connectivity to be present in order to be applied (which is guaranteed for all resolutions except the coarsest), Λ_k and Φ_k support extraordinary vertices, unlike [8]. Furthermore, neither [7] nor [8] provide smooth reverse dual schemes, which are supported in our work via Φ_k.

A notable consequence of defining the subdivision, reverse subdivision, and smooth reverse schemes in terms of Λ_k and Φ_k is that the smooth reverse scheme can be used to cancel out Λ_k^{-1} and Φ_k^{-1} in the reverse subdivisions. As a result, we can convert a given reverse subdivision scheme into another.

Consider, for instance, a primal subdivision and reverse subdivision scheme defined using two averaging steps: $P_{Primal} = \Lambda_1 \circ \Lambda_0 \circ I_P$ and $\hat{P}_{Primal} = \hat{I}_P \circ \Lambda_0^{-1} \circ \Lambda_1^{-1}$. If we apply the smoothing operation $\Delta = \Lambda_1 \circ \Lambda_0$ after the reverse subdivision, then consecutive reverse subdivision operations reduce to $\hat{I}_P$

$$\Delta \circ \hat{P}_{Primal} \circ \Delta \circ \hat{P}_{Primal} = \Lambda_1 \circ \Lambda_0 \circ \hat{I}_P \circ \hat{I}_P \circ \Lambda_0^{-1} \circ \Lambda_1^{-1}.$$

The forward scheme, consequently, becomes

$$P_{Primal} \circ \Delta^{-1} \circ P_{Primal} \circ \Delta^{-1} = \Lambda_1 \circ \Lambda_0 \circ I_P \circ I_P \circ \Lambda_0^{-1} \circ \Lambda_1^{-1}.$$

4.2 Non-Uniform and Adaptive Primal Subdivision

In the unmodified RIAS framework, when we apply the averaging step Λ_k, a single smoothing weight s_k is used across the entire mesh M. However, this need not be the case, and we can define a non-uniform subdivision quite simply by varying the smoothing weight across the mesh. Using this approach, different subdivision rules can be blended together by blending different smoothing weights, without the need to explicitly define new vertex masks.

In order to produce non-uniform primal subdivision, we associate a set of smoothing weights $S^v = \{s_0^v, s_1^v, \ldots, s_{m-1}^v\}$ with each vertex v in M (see Fig. 5). As we may not wish to have all averaging steps act on a given vertex (m is constant
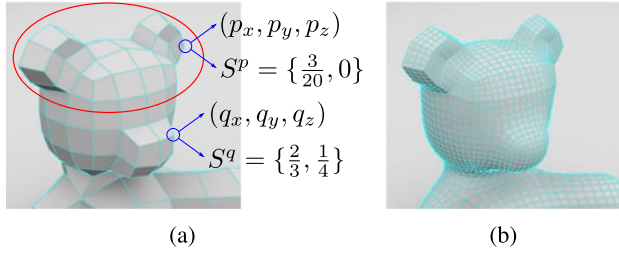
Fig. 5. Non-uniform subdivision. (a) A set of weights is assigned to each vertex: the circled region receives weights $\{\frac{3}{20}, 0\}$ while other vertices receive the weights $\{\frac{2}{3}, \frac{1}{4}\}$. (b) The non-uniformly subdivided mesh.

TABLE 2
Top Five Weight Sets Ranked by Performance on
Two Metrics: Average Detail Length Minimization
and Shape Error Minimization

Rank	Detail Length		Shape Error	
	Weights	Length $\times 10^{-2}$	Weights	Error $\times 10^{-2}$
1	$\{\frac{1}{5}\}$	1.377	$\{\frac{1}{20}\}$	2.583
2	$\{\frac{1}{4}\}$	1.383	$\{\frac{1}{10}\}$	2.603
3	$\{\frac{3}{20}\}$	1.390	$\{\frac{3}{20}\}$	2.640
4	$\{\frac{1}{10}\}$	1.413	$\{\frac{1}{5}\}$	2.701
5	$\{\frac{3}{10}\}$	1.419	$\{\frac{1}{20}, \frac{1}{20}\}$	2.747

across the mesh), the weight corresponding to a given averaging step can be set to zero in order to eliminate its effect. For any operation that acts on the vertex v, rather than using s_k we instead use s_k^v.

A simple way to implement this method is to define a *weight mesh* W with topology identical to M but with smoothing weights (s_0^v, s_1^v, s_2^v) in place of vertex coordinates (x, y, z). For $k = 0, 1, 2$, the smoothing weight s_k^v for a given vertex v can be obtained by checking the coordinates of the corresponding vertex in W. Weights for subdivided versions of M can be obtained by subdividing W (note that the subdivision connectivity must be identical across both meshes, but otherwise any arbitrary subdivision method can be used on W — even an unmodified RIAS). We use I_P to subdivide W, which interpolates the original weights s_k^v.

Applications of this method include targeting smooth reverse subdivision to specific parts of the mesh and defining transition regions between areas of different resolution, for instance in a geometry clip-map [25] or in an adaptive subdivision [26]. As an example, we can extract a patch from the mesh, non-uniformly subdivide it such that the boundary shape remains unchanged ($S^v = \{0, 0, \ldots, 0\}$ for v on the boundary), and visualize the high-resolution patch in place of the coarse version. Unlike other adaptive methods, the subdivision connectivity of the mesh does not need to be altered, and thereby does not lose the benefits of multiresolution. Furthermore, by virtue of RIAS's reversibility, the impacts of the high-resolution edits can be reversed and incorporated back into the coarse mesh.

5 SMOOTHING WEIGHTS

Given that our multiresolution schemes depend on sets of smoothing weights S, an important question that arises is that of which weight values to choose. The answer to this question depends on several factors, such as whether or not a B-Spline surface is desired or if the scheme should satisfy some particular application requirement(s).

In this section, we examine the question of smoothing weight selection in detail. First we present a weight recipe that produces B-Spline multiresolution, and second we note how smoothing weights can be tailored to different applications.

5.1 Weights for B-Spline Multiresolution

While [6] notes that the set of weights $S = \{\frac{1}{2}\}$ produces degree 2 and 3 B-Spline multiresolution in the dual and primal case, respectively, no other weights for B-Spline schemes were presented in that work. Here, we outline a recipe for the weights that allow both our methods and the methods of [6]

to produce B-Spline schemes of any degree. This recipe was found by running a brute force Octave script to find appropriate weights for different degrees, and then generalizing from the observed pattern.

For a degree $2m$ or $2m+1$ B-Spline scheme, the m weights that produce that scheme can be found as the set of odd numbers from $2m-1$ to 1 divided by the set of natural numbers from $2m$ to $m+1$, i.e.,

$$ S = \left\{ \frac{2m-1}{2m}, \frac{2m-3}{2m-1}, \ldots, \frac{3}{m+2}, \frac{1}{m+1} \right\}. $$

Hence, $S = \{\frac{1}{2}\}$ produces degree 2 and 3 B-Spline schemes (Doo-Sabin and a variant of Catmull-Clark, respectively), $S = \{\frac{3}{4}, \frac{1}{3}\}$ produces degree 4 and 5 B-Spline schemes, $S = \{\frac{5}{6}, \frac{3}{5}, \frac{1}{4}\}$ produces degree 6 and 7 B-Spline schemes, etc.

The production of B-Spline schemes using this recipe can be verified on a case-by-case basis by deriving the masks for regular vertices (see supplemental material, available online), and has been verified for all degrees up to 20 using Octave. Though a formal proof was attempted, the change of weights between degrees renders induction ineffective and complicates attempts to perform this verification in the general case. Approaches based on generating functions are hampered by the non-uniform behaviours of Λ_k and Φ_k, which prevents them from being modeled as convolution operators and applied to generating functions using polynomial multiplication.

5.2 Tailoring Smoothing Weights

Though B-Spline surfaces are well-known and widely used, they and the smoothing weights that produce them are not equally well-suited to every task. We have noted, for instance, in Section 4.1 that shape exaggerations in reverse subdivided meshes can encourage the use of scaled down weights.

As different applications will have differing objectives with regards to multiscale surface representations, it bears noting that the smoothing weights of the RIAS framework can be selected with an eye towards satisfying these objectives.

For example, we can attempt to select whichever weights produce the best mesh compression, measured by either the average length of detail vectors or the average error between the coarsened and original mesh shapes. Table 2 shows a sample ranking of weight sets ranked according to these metrics. Results are for the high resolution Muffin mesh shown in Fig. 6a, tested using uniformly sampled weight sets of the form $S = \{s_0, s_1\}$ ($s_0 \neq 0$).
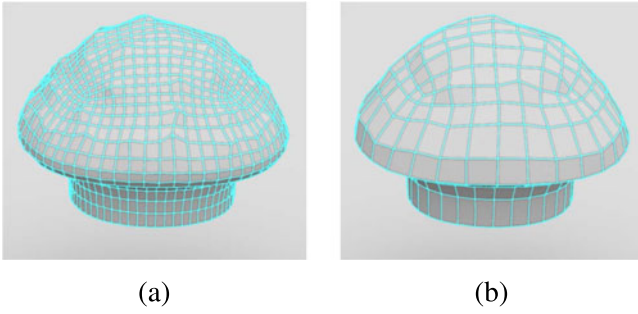
Fig. 6. (a) High resolution Muffin mesh. (b) Muffin mesh after 2 application of P_{Primal} with $S = \{\frac{3}{20}\}$.

For this mesh, the weight set $S = \{\frac{3}{20}\}$ appears to perform admirably well, being in the top three rankings of both metrics. This may not be the case for all meshes, however. Using RIAS, we can obtain schemes that maximize metrics for different applications and on different meshes.

6 CONTINUITY AT EXTRAORDINARY VERTICES

Similarly to the works of [2], [3], the limit subdivision surfaces resulting from our methods exhibit C^d continuity at regular vertices (when using the $\lfloor\frac{d}{2}\rfloor$ weights as described in Section 5.1). This is because, in regular cases, the vertex masks resulting from these weights reduce to the tensor product generalization of the vertex masks from [6], which produce B-Spline subdivision schemes (refer to supplemental material, available online).

However, analyzing the continuity at extraordinary vertices is not quite so easy. For each extraordinary vertex or face, we can consider an invariant local neighbourhood around the face or vertex and a local subdivision matrix P_L that transforms that neighbourhood into a corresponding neighbourhood at the next resolution (see [16, Chapter 8]). The eigenvalues $\lambda_0, \lambda_1, \lambda_2, \ldots$ and characteristic map ψ resulting from P_L determine the continuity of the subdivision scheme at those extraordinary vertices or faces.

As per [16, Chapter 8], we order the eigenvalues by decreasing magnitude, i.e., $\lambda_0 = 1 > |\lambda_1| = |\lambda_2| \geq |\lambda_i|$, for all $i > 2$. The subdominant eigenvectors e_1 and e_2 corresponding to λ_1 and λ_2 can be used to determine the characteristic map $\psi = (\psi_1, \psi_2)$, which parametrizes the limit subdivision surface and must be "regular" (one-to-one and onto). If ψ is regular and $|\lambda_2| > |\lambda_i|$ for all $i > 2$, then the subdivision scheme associated with P_L converges to a C^1 limit surface.

While we have not formally analyzed the characteristic maps ψ for our schemes, the characteristic meshes (resulting
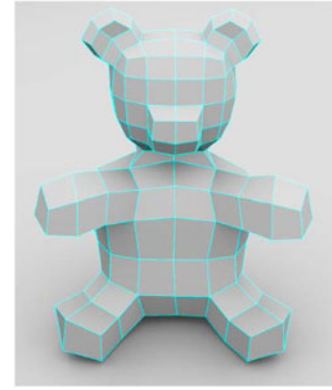

Fig. 8. Coarse teddy mesh.

from forming the invariant local neighbourhood using entries of e_1 and e_2) suggest that each ψ is regular (see Fig. 7). The planar spread of the meshes in addition to the lack of self-intersections supports this claim. We have additionally verified (for all B-Spline schemes from degree 2 to 15, up to valence 20) that the correct eigenvalue structure holds for C^1 continuity, suggesting that our schemes have C^1 continuity at extraordinary vertices.

We have not analyzed the smoothness of the blended subdivision schemes used to construct non-uniform and adaptive subdivision, and suspect that (when using I_P to combine smoothing weights) the smoothness of the original/parent schemes will not carry over to the blended scheme. How to combine the smoothing weights such that smoothness is preserved in the blending scheme remains an open question for future work.

7 RESULTS

We present in this section some results from applying our multiresolution schemes on meshes.

We have subdivided several meshes using different B-Spline subdivision schemes. Figs. 9, 10, 11, and 12 illustrate our subdivision and reverse subdivision schemes (from degree 2 to degree 5) on the coarse Teddy mesh shown in Fig. 8. In each case, we introduce perturbations into the fine mesh before reverse subdivision in order to examine the effect of shape exaggerations. We have found that shape exaggerations are not especially problematic for the lower degree schemes (i.e., 2 and 3), and can be minimized using smooth reverse approaches in the case of higher degree schemes.
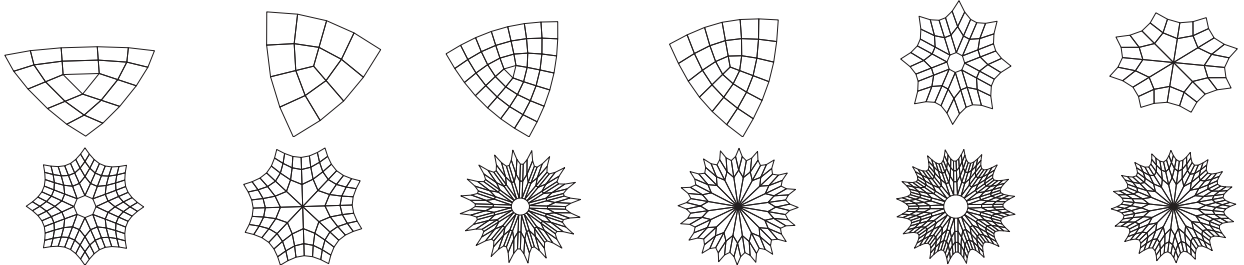

Fig. 7. Characteristic meshes for our primal and dual schemes. Meshes shown are for extraordinary valences 3, 8, and 20 from degree 2 to degree 5.
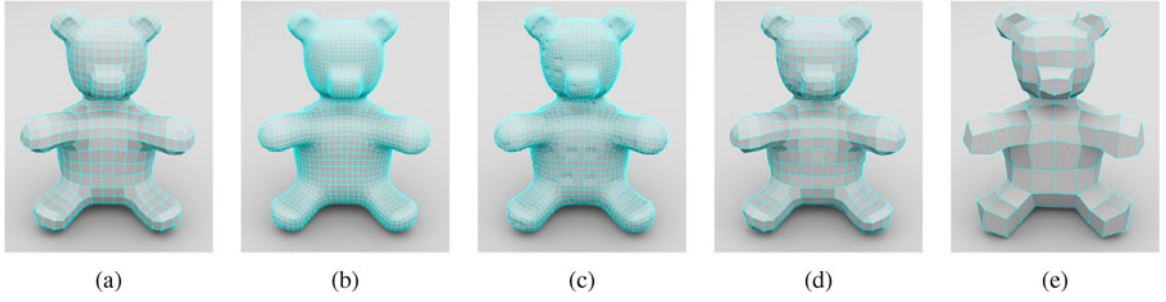
Fig. 9. (a), (b) Dual subdivision applied on Teddy using weights $S = \{\frac{1}{2}\}$. (c) High resolution Teddy with perturbations. (d), (e) Reverse dual subdivision on the mesh in (c) using weights $S = \{\frac{1}{2}\}$.
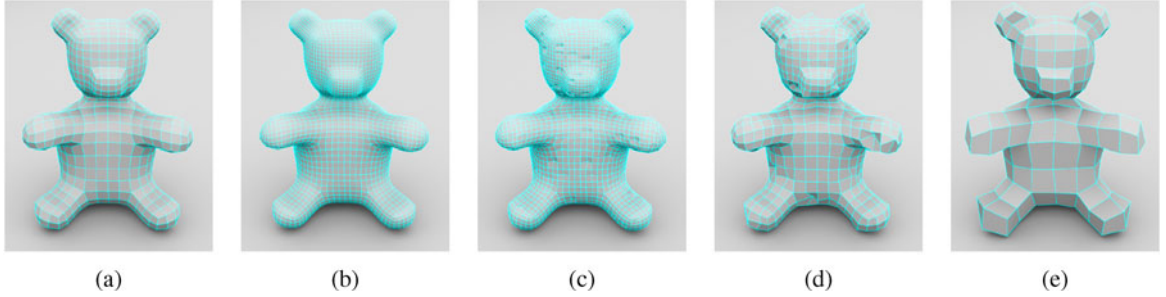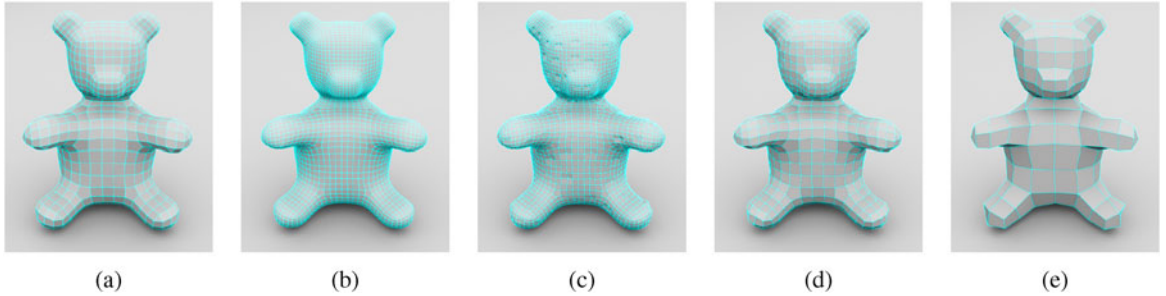

Fig. 10. (a), (b) Primal subdivision applied on Teddy using weights $S = \{\frac{1}{2}\}$. (c) High resolution Teddy with perturbations. (d), (e) Reverse primal subdivision on the mesh in (c) using weights $S = \{\frac{1}{2}\}$.

Fig. 11. (a), (b) Dual subdivision applied on Teddy using weights $S = \{\frac{3}{4}, \frac{1}{3}\}$. (c) High resolution Teddy with perturbations. (d), (e) Reverse dual subdivision on the mesh in (c) using weights $S = \{\frac{3}{8}, \frac{1}{6}\}$.
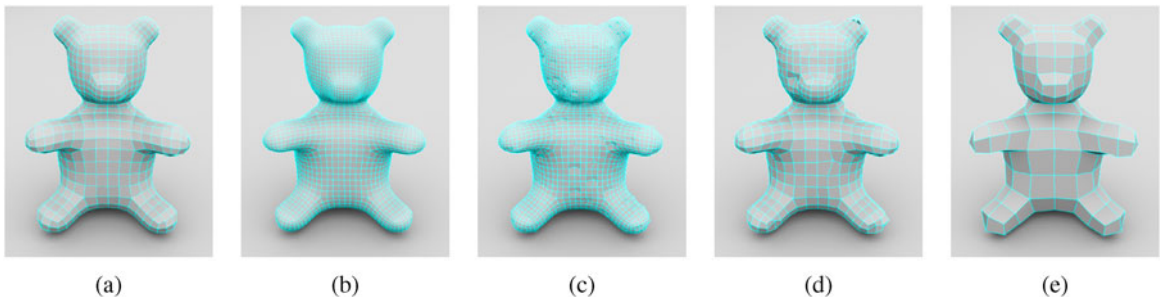

Fig. 12. (a), (b) Primal subdivision applied on Teddy using weights $S = \{\frac{3}{4}, \frac{1}{3}\}$. (c) High resolution Teddy with perturbations. (d), (e) Reverse primal subdivision on the mesh in (c) using weights $S = \{\frac{3}{8}, \frac{1}{6}\}$.

Figs. 9 and 10 feature, respectively, our dual and primal schemes (using weights $S = \{\frac{1}{2}\}$), and correspond to Doo-Sabin subdivision and our variant of Catmull-Clark subdivision. Figs. 11 and 12 feature, respectively, our dual and primal subdivision schemes (using weights $S = \{\frac{3}{4}, \frac{1}{3}\}$) and reverse subdivision schemes (using scaled weights $S = \{\frac{3}{8}, \frac{1}{6}\}$ to

reduce shape exaggerations). See Fig. 13 for results from a non B-Spline subdivision, which uses the weights $S = \{\frac{2}{3}, \frac{1}{4}\}$ reported in [6]. The RIAS framework allows us to use simple operations in order to generate highly smooth subdivision surfaces and, unlike the classical approach [2], [3], to reverse the surfaces back to a coarser resolution.
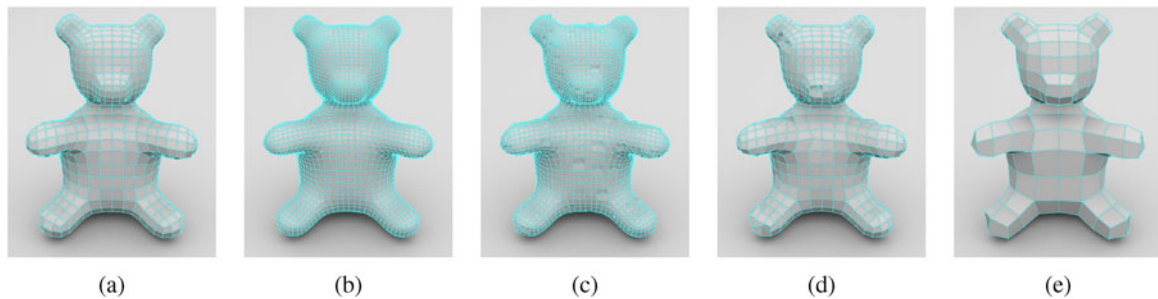
Fig. 13. Non B-Spline subdivision example. (a), (b) Primal subdivision applied on Teddy using weights $S = \{\frac{2}{3}, \frac{1}{4}\}$. (c) High resolution Teddy with perturbations. (d), (e) Reverse primal subdivision on the mesh in (c) using weights $S = \{\frac{2}{6}, \frac{1}{8}\}$.
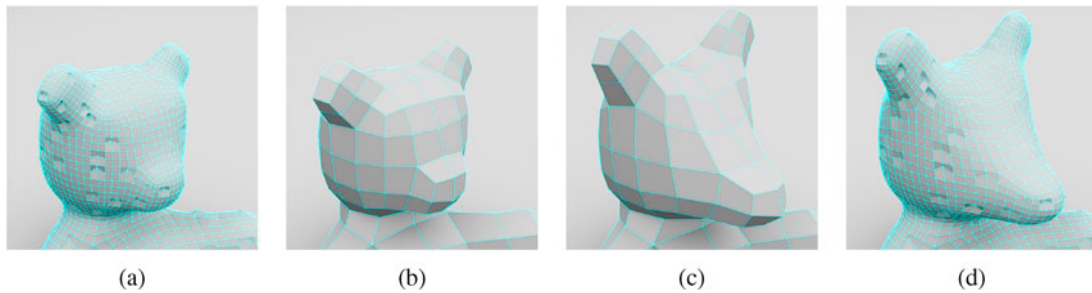

Fig. 14. Multiresolution editing example. (a) High resolution Teddy mesh with perturbations. (b) Reverse subdivided coarse mesh. (c) The coarse mesh is edited to resemble a fox. (d) High resolution "Fox" mesh with details preserved.

Examples of multiresolution editing are shown in Figs. 14 and 15c. While shape exaggerations typically cause high degree multiresolution editing to be challenging to use (see, e.g., Fig. 15a), our framework supports means by which a coarse version of a high resolution mesh can be edited intuitively and without loss of detail. Fig. 14 uses scaled weights (i.e., $\{\frac{3}{8}, \frac{1}{6}\}$) in a degree 5 reverse scheme, while Fig. 15c utilizes a degree 3 smooth reverse framework (see Section 4.1).
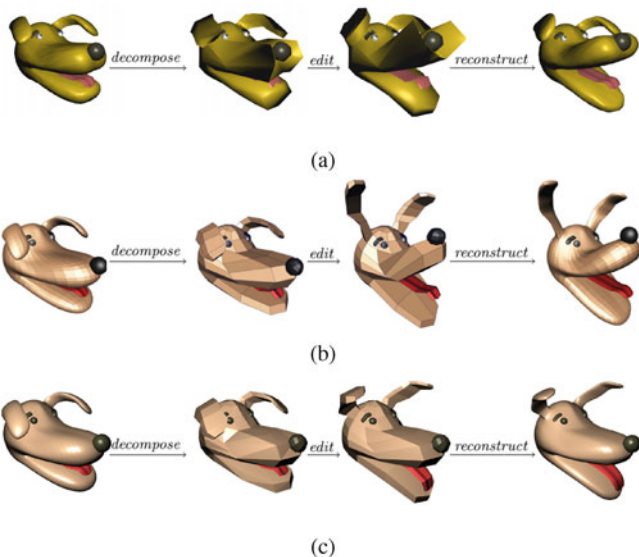

Fig. 15. (a) Standard reverse subdivision approaches, as in [10], cause shape exaggerations that make multiscale editing less intuitive. (b) Smooth reverse subdivision, as in [7], reduces such exaggerations. (c) Result from our smooth reverse framework after decomposing a high resolution dog mesh three times. We used a weight of $\frac{1}{2}$ for the reverse scheme $\hat{P}_{Primal}$ and a weight of $\frac{1}{4}$ for the smoothing step Δ. Results appear comparable to [7] but require no additional implementation effort, and unlike [7], [8] do not suffer from overrepresentation or topological restrictions. Figures from [10] and [7] used with permission.
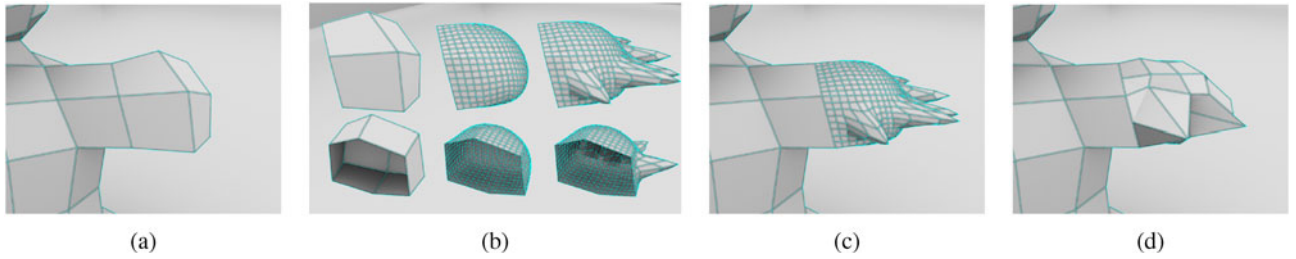
As both approaches are constructed using the averaging steps required by the base framework, no additional implementation is required to achieve smooth reverse subdivision. The results of our smooth reverse framework appear to be comparable to those of [7] and [8], but our method does not suffer from overrepresentation or from topological restrictions.

An example of non-uniform and adaptive subdivision can be found in Fig. 16. By using a linear shape-preserving subdivision on the boundary of an extracted mesh patch, we can create a high-resolution version of the patch that fits the remaining coarse mesh. The framework's reversibility allows the patch to be reverse subdivided in case we wish to re-integrate the patch with the rest of the mesh.

Fig. 17 presents reflection lines on the Teddy mesh after four applications of various types of subdivision. Images were created using the Zebra line functionality in Rhino. Our results appear comparable to the work of Stam [2], with curvature divergence similarly worsening with degree at valence 3 extraordinary vertices but improving with degree at higher valence vertices. As expected, the smoothness at regular vertices improves with degree. How to improve the behaviour of valence 3 vertices as degree increases is an interesting future direction for both RIAS and the classical approach.

In Table 3, we present a runtime performance comparison of RIAS subdivision with the classical approach (as described in [2]). All tests were run on a 64-bit Windows 10 machine with an Intel Core i7-6700k CPU and 16 GB of RAM. Both methods were implemented as serial algorithms, and executed 1000 times each on a regular grid of 200×200 points.

Though we can expect RIAS and the classical approach to produce the same types of B-Spline surfaces given the same number of averaging steps (i.e., degree $2m$ or $2m + 1$ with C^{2m-1} or C^{2m} continuity at regular vertices and C^1 at extraordinary vertices), there are significant differences in the approaches that make RIAS more amenable to performance optimizations.

Fig. 16. Non-uniform/adaptive subdivision example. (a) Coarse Teddy mesh. (b) Teddy's hand is extracted, subdivided (3 applications of P_{Primal} with $S = \{\frac{1}{2}\}$ on interior vertices, $S = \{0\}$ on boundary vertices), and edited. (c) Coarse Teddy with a high-resolution claw. (d) Reverse subdivision can be applied to the extracted mesh (2 applications of $\hat{P}_{Primal}$ with $S = \{\frac{1}{4}\}$ on interior vertices, $S = \{0\}$ on boundary vertices).
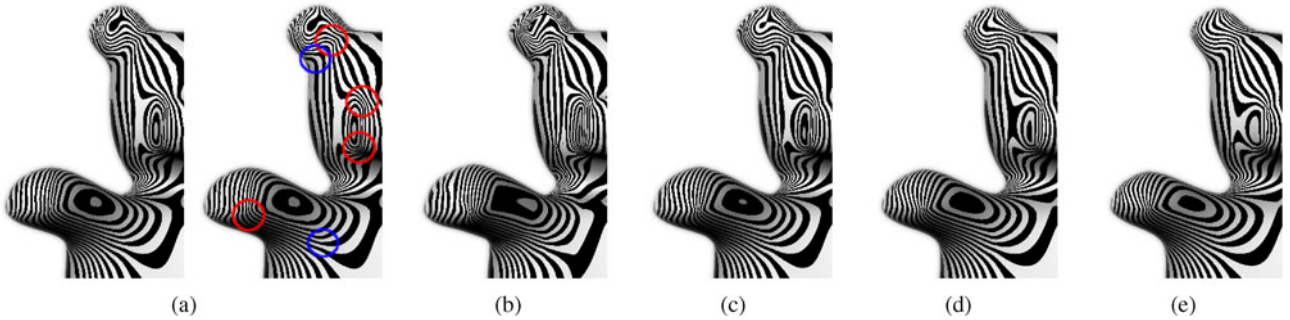

Fig. 17. Reflection lines on Teddy after (a) Catmull-Clark subdivision; (b), (c) dual and primal subdivision, respectively, with $S = \{\frac{1}{2}\}$; (d), (e) dual and primal subdivision, respectively, with $S = \{\frac{3}{4}, \frac{1}{3}\}$. Extraordinary vertex neighbourhoods are highlighted in (a) using red circles (valence 3) and blue circles (valence 5).

For instance, whenever a vertex is moved in the classical approach, its original position must be cached for use in calculating its neighbours' positions. This temporarily doubles the footprint of the mesh in shared memory, though the runtime impact can be mitigated using a double-buffering approach. Such caching is not required by RIAS's primal step, and in the dual case can be performed per shrinking face in local memory. Furthermore, each averaging step in RIAS is less expensive. In the primal case, 25 percent of the vertices are fixed at a time and require no calculations, while calculations for edge-vertices (which make up 50 percent of the vertices) require only two vertex positions from the one-ring. In the dual case, a vertex's position is calculated using the vertices from a single incident face, rather than all incident faces. As a result, RIAS subdivision appears to perform 40 percent better in the dual case and 60 percent better in the primal case (before parallelization).

8 CONCLUSIONS AND FUTURE WORK

We have presented two algorithms each for subdividing and reverse subdividing a polygonal mesh based on repeated applications of invertible averaging operators. Together, they establish a multiresolution framework on general topology surfaces that can achieve C^{td} continuity at regular vertices and appear to have C^1 continuity at extraordinary vertices.

Our construction unifies several existing concepts into a single comprehensive framework for the manipulation of a surface's resolution, tying arbitrarily smooth subdivision surfaces together with smooth reverse subdivision, non-uniform subdivision, and multiresolution representations. All operations, by design, can be implemented in terms of the one-ring of individual vertices and can be configured to meet different objectives. Furthermore, the averaging steps themselves are amenable to parallelization with minimal overhead, and are therefore capable of significant performance improvements over existing methods that generate arbitrarily smooth subdivision surfaces.

For future work, it would be interesting to investigate the adjustment of averaging masks in extraordinary cases to produce continuity beyond C^1. Other work could explore different approaches to blending smoothing weight sets, such that a blended subdivision scheme is made at least as smooth as one of its parent schemes.

Challenges remain in deriving closed-form formulae for our vertex masks and in proving that our smoothing weight recipe produces B-Spline schemes in general. Changes in weight sets between degrees reduce the utility of induction approaches, and the derivation of generating functions is complicated by the fact that our averaging operators cannot be modeled as convolutions. Addressing either of these issues could form the basis for future work.

TABLE 3
Average Runtime Performance of RIAS Subdivision and the
Classical Approach, in Milliseconds (ms)

Number of Averaging Steps	Dual Schemes		Primal Schemes	
	RIAS	[2], [3]	RIAS	[2], [3]
1	39.741	67.743	36.324	86.364
2	74.741	126.776	66.015	147.681
3	111.063	187.493	82.323	205.576
4	145.267	245.178	104.168	262.788

REFERENCES

[1] J. M. Lane and R. F. Riesenfeld, "A theoretical development for the computer generation and display of piecewise polynomial surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 1, pp. 35–46, Jan. 1980.

[2] J. Stam, "On subdivision schemes generalizing uniform B-Spline surfaces of arbitrary degree," *Comput. Aided Geometric Des.*, vol. 18, no. 5, pp. 383–396, 2001.

[3] D. Zorin and P. Schröder, "A unified framework for primal/dual quadrilateral subdivision schemes," *Comput. Aided Geometric Des.*, vol. 18, no. 5, pp. 429–454, 2001.

[4] E. Catmull and J. Clark, "Recursively generated B-spline surfaces on arbitrary topological meshes," *Comput.-Aided Des.*, vol. 10, no. 6, pp. 350–355, 1978.

[5] D. Doo and M. Sabin, "Behaviour of recursive division surfaces near extraordinary points," *Comput.-Aided Des.*, vol. 10, no. 6, pp. 356–360, 1978.

[6] T. F. Alderson, A. Mahdavi-Amiri, and F. F. Samavati, "Multiresolution on spherical curves," *Graphical Models*, vol. 86, pp. 13–24, 2016.

[7] J. Sadeghi and F. F. Samavati, "Smooth reverse Loop and Catmull-Clark subdivision," *Graphical Models*, vol. 73, no. 5, pp. 202–217, 2011.

[8] J. Sadeghi and F. F. Samavati, "Local fairing with local inverse," in *Proc. Graph. Interface Conf.*, 2013, pp. 117–124.

[9] L. J. Olsen, F. F. Samavati, and R. H. Bartels, "Multiresolution for curves and surfaces based on constraining wavelets," *Comput. Graph.*, vol. 31, no. 3, pp. 449–462, 2007.

[10] L. J. Olsen and F. F. Samavati, "A discrete approach to multiresolution curves and surfaces," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2008, pp. 468–477.

[11] C. Loop, "Smooth subdivision surfaces based on triangles," Master's thesis, Dept. Math., Univ. Utah, Salt Lake City UT, 1987. [Online]. Available: https://www.microsoft.com/en-us/research/publication/smooth-subdivision-surfaces-based-on-triangles/

[12] J. Peters and U. Reif, "The simplest subdivision scheme for smoothing polyhedra," *ACM Trans. Graph.*, vol. 16, no. 4, pp. 420–431, 1997.

[13] T. J. Cashman, "Beyond Catmull-Clark? A survey of advances in subdivision surface methods," *Comput. Graph. Forum*, vol. 31, no. 1, pp. 42–61, 2012.

[14] U. Reif, "A unified approach to subdivision algorithms near extraordinary points," *Comput.-Aided Geometric Des.*, vol. 12, no. 2, pp. 153–174, 1995.

[15] D. Zorin, "Stationary subdivision and multiresolution surface representations," PhD dissertation, Dept. Comput. Sci., California Inst. Technol. Pasadena CA, 1997. [Online]. Available: https://resolver.caltech.edu/CaltechCSTR:1997.cs-tr-97-32

[16] J. Warren and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*. San Francisco, CA, USA: Morgan Kaufmann, 2002.

[17] J. Peters and U. Reif, "Analysis of algorithms generalizing B-spline subdivision," *SIAM J. Numer. Anal.*, vol. 35, no. 2, pp. 728–748, 1998.

[18] F. F. Samavati and R. H. Bartels, "Multiresolution curve and surface representation: Reversing subdivision rules by least-squares data fitting," *Comput. Graph. Forum*, vol. 18, no. 2, pp. 97–120, 1999.

[19] S. Lanquetin and M. Neveu, "Reverse Catmull-Clark subdivision," in *Proc. 14th Int. Conf. Central Eur. Comput. Graph. Vis. Comput. Vis.*, 2006, pp. 319–326.

[20] F. F. Samavati, N. Mahdavi-Amiri, and R. H. Bartels, "Multiresolution surfaces having arbitrary topologies by a reverse Doo subdivision method," *Comput. Graph. Forum*, vol. 21, no. 2, pp. 121–134, 2002.

[21] H. Guo, Y. Guan, M. Liu, and K. Qin, "Biorthogonal wavelet transforms and applications based on generalized progressive Catmull-Clark subdivision with shape control," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 7, pp. 2392–2403, Jul. 2019.

[22] R. H. Bartels and F. F. Samavati, "Reversing subdivision rules: Local linear conditions and observations on inner products," *J. Comput. Appl. Math.*, vol. 119, no. 1/2, pp. 29–67, 2000.

[23] T. F. Alderson and F. F. Samavati, "Multiscale NURBS curves on the sphere and ellipsoid," *Comput. Graph.*, vol. 82, pp. 243–249, 2019.

[24] E. J. Stollnitz, T. D. Derose, and D. H. Salesin, *Wavelets for Computer Graphics: Theory and Applications*. San Francisco, CA, USA: Morgan Kaufmann, 1996.

[25] F. Losasso and H. Hoppe, "Geometry clipmaps: Terrain rendering using nested regular grids," *ACM Trans. Graph.*, vol. 23, pp. 769–776, 2004.

[26] H.-R. Pakdel and F. F. Samavati, "Incremental subdivision for triangle meshes," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 1, pp. 80–92, 2007.

Troy Alderson received the BSc's degree in computer science and general mathematics and the MSc degree in computer science, both from the University of Calgary, Calgary, Canada, in 2010 and 2014. He is currently working toward the PhD degree with the Department of Computer Science, University of Calgary, Calgary, Canada. His current research interest focuses on developing unified frameworks for subdivision and multiresolution for various types of geometric shapes.

Ali Mahdavi-Amiri received the PhD degree from the University of Calgary, Calgary, Canada, in 2015. He is currently a University research associate with GrUVi Lab, Simon Fraser University, Canada. His research interests include geometric modeling and computational creativity, design, and fabrication. He has received several outstanding grants and awards including NSERC postdoctoral fellowship, J.B. Hyne Research Innovation Award, and Michael A. J. Sweeney Award for the best paper at Graphics Interface, 2013.

Faramarz Samavati is currently a professor with the Department of Computer Science, University of Calgary. His research interests include computer graphics, visualization, 3D imaging, and geometric modeling. In the past seven years, he has received seven best paper awards, Digital Alberta Award, Great Supervisor Award, and University of Calgary Peak Award, which honors his contribution to the development of new technologies and innovations.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.