Computers & Graphics 82 (2019) 243–249

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on SMI 2019

Multiscale NURBS curves on the sphere and ellipsoid

Troy Alderson*, Faramarz Samavati

Department of Computer Science, University of Calgary, Calgary, AB, Canada

ARTICLE INFO

Article history:
Received 20 March 2019
Revised 25 May 2019
Accepted 27 May 2019
Available online 3 June 2019

Keywords:
NURBS
Subdivision
Reverse subdivision
Multiresolution
Spherical space

ABSTRACT

In this paper, we introduce a framework that allows NURBS subdivision curves to be defined on the sphere and ellipsoid in a multiscale manner. This is achieved via modification of a repeated invertible averaging (RIA) framework for spherical B-Spline curves, which is constructed in terms of spherical linear interpolations. By incorporating vertex weights into the interpolation parameters of individual operations, and by generalizing the linear interpolations to other manifolds, we can define multiscale NURBS on several types of surfaces. We explore an application to the multiscale representation of geospatial vector data and present an optimization method that automatically assigns NURBS vertex weights to curve vertices.

1. Introduction

When it comes to curve design, freeform curves such as Bezier, B-Spline, and NURBS curves are well-understood and widely utilized. NURBS curves are particularly notable; through the use of vertex weights, they feature improved control over other types of curves and the ability to represent conic sections exactly. As a result, NURBS curves have become ubiquitous in CAD/CAM as well as in numerous industry standards [1].

The popularity of each type of freeform curve has naturally inspired a variety of works that generalize these curves to other spaces. The surface of a sphere has been an especially prominent target for such generalizations, as curves on the sphere have several applications in geospatial vector representation [2], the creation of rotation splines for key-frame animation [3], and the generation of tool paths for five-axis machining [4]. The mathematical elegance of the sphere itself has additionally provided fertile ground for various artistic endeavours, such as the creation of Islamic star patterns and Persian floral patterns (see, e.g., [5,6]).

However, many such works are unable to represent their spherical curves in a multiscale manner (for the purposes of, e.g., editing, data transmission, or level-of-detail rendering). In order to address this, the work of Alderson et al. [2] proposed a methodology by which B-Spline curves on the surface of a sphere could be represented in a multiscale manner. The method, which we refer to as *repeated invertible averaging (RIA)*, is based on a modified version of the Lane-Riesenfeld algorithm [7] and uses a sequence of *SLERP* (spherical linear interpolation; see [8]) operations to perform its calculations. Via replacement of *SLERP* with an analogous interpolation method defined along geodesic lines (say, $Interp(p, q, u)$), the method can be easily generalized to other manifold surfaces.

In order to bring the same freedom of representation to spherical NURBS curves, we introduce a modification to RIA that allows it to represent NURBS curves at multiple scales on the surface of the sphere (and any other manifold supported by RIA). In essence, this modification converts the interpolation parameter u into a function of the NURBS vertex weights, based on the observed effect of the geometric interpretation of NURBS curves (see [1]) on individual interpolation operations. We demonstrate use of the modified framework in the creation of subdivided and reverse subdivided spherical NURBS curves, and explore an application of this framework to representing geospatial vector data. We additionally utilize an implementation of the algorithms described in [9] to create multiscale NURBS curves on ellipsoids of revolution.

The paper is organized as follows. In Section 2, we review a selection of related works. Sections 3 and 4, respectively, provide overviews of the RIA framework and the geometric interpretation of NURBS curves. Our modification to the framework is described in Section 5, followed by a discussion on geospatial vector representation in Section 6. Results may be found in Section 7 and conclusions in Section 8.

2. Related work

Shoemake's influential work [3] performed the first generalization of Bezier curves to the surface of a sphere, for the purposes of creating rotation curves for smooth animations. This was achieved by utilizing *SLERP* in place of linear interpolations within the well-known de Casteljau algorithm.

* Corresponding author.
E-mail addresses: tfalders@ucalgary.ca (T. Alderson), samavati@ucalgary.ca (F. Samavati).

https://doi.org/10.1016/j.cag.2019.05.029
0097-8493/© 2019 Published by Elsevier Ltd.

An alternative approach was proposed by Buss and Fillmore in [10], where spherical weighted averages were formulated as least squares minimization problems rather than as sequences of *SLERP* operations. Spherical versions of the parametric Bezier and B-Spline curve equations were defined using this new average. Interestingly, the peculiar nature of weighted averages on the sphere suggests that the Bezier curves of Shoemake [3] and Buss and Fillmore [10] do not perfectly coincide in general. However, both approaches are consistent with Euclidean formulations and should converge as the radius of the sphere approaches infinity (i.e., the surface becomes planar).

Later works adopted approaches similar to Shoemake's in order to transfer other types of curves onto the sphere. Generalizations of de Boor's algorithm using *SLERP* were used to produce spherical B-Spline curves in [11] and spherical NURBS curves in [4].

The work of Schaefer and Goldman [11] additionally explored the creation of subdivision curves on the surface of the sphere by generalizing the Lane-Riesenfeld algorithm [7]. The algorithm, which subdivides a given set of points in such a way that a B-Spline subdivision curve of any desired degree is produced, consists of two simple operations. The first is a very simple initial subdivision, followed by several averaging steps that move each point to the (geodesic) midpoint it forms with its consecutive neighbour.

The Lane-Riesenfeld averaging step, however, is not invertible, and so a similar approach to reverse the subdivision process did not exist for many years. As reverse subdivision is a key component of subdivision-based multiresolution/multiscale frameworks (see, e.g., [12,13]), this meant that the resulting curves could not be represented in a multiscale manner.

In [2], a modified version of the Lane-Riesenfeld algorithm was introduced whose averaging steps were invertible. The resulting repeated invertible averaging (RIA) approach could be generalized to spheres, and was used to create multiscale B-Spline curves on the surface of a sphere. Using algorithms for geodesics on the ellipsoid [9,14,15], the method can additionally be used on ellipsoids of revolution. We introduce a modification to this method that extends it to the multiscale representation of NURBS.

3. Review of RIA

The repeated invertible averaging, or RIA, framework is a framework for the multiscale representation of a class of curves (B-Splines included[1]) in Euclidean and various non-Euclidean spaces. All aspects of the multiresolution transformations – such as subdivision and reverse subdivision – are defined in terms of sequences of linear interpolations, which allows the framework to be generalized to any space for which a linear interpolation can be defined (e.g., spheres via *SLERP*).

Given a vector of 3D points $\boldsymbol{p} = [p_0, p_1, \ldots, p_{n-1}]^T$, $p_i \in \mathbb{R}^3$, subdivision via RIA is accomplished by applying an initial basic subdivision to the points, followed by one or more applications of an invertible averaging (or smoothing) step. Both dual- and primal-type subdivisions are supported by the framework, via two different initial subdivision and averaging steps. A user-provided set of smoothing weights $S = \{s_0, s_1, \ldots, s_{m-1}\}$, $s_i \neq 0, 1$, defines the number of averaging steps applied and the strength of the smoothing performed by each step. Example sets include $S = \{\frac{1}{2}\}$, which gives degree 2 and 3 B-Spline subdivision, and $S = \{\frac{3}{4}, \frac{1}{3}\}$, which gives degree 4 and 5 B-Spline subdivision.

Let $Interp(p, q, u)$ denote a linear interpolation method in the desired space, where $p, q \in \mathbb{R}^3$ are points and $u \in \mathbb{R}$ is the interpolation parameter. Notable candidates include spherical linear interpolation [3]

$$SLERP(p, q, u) = \frac{\sin[(1-u)\theta]}{\sin(\theta)} p + \frac{\sin(u\theta)}{\sin(\theta)} q,$$

where θ is the angle between p and q, and Euclidean linear interpolation

$$LERP(p, q, u) = (1-u) \cdot p + u \cdot q.$$

[1] As with the Lane-Riesenfeld algorithm, subdividing with RIA has the same effect as inserting knots into the B-Spline's knot vector, where knots are inserted at the midpoint of each knot interval.

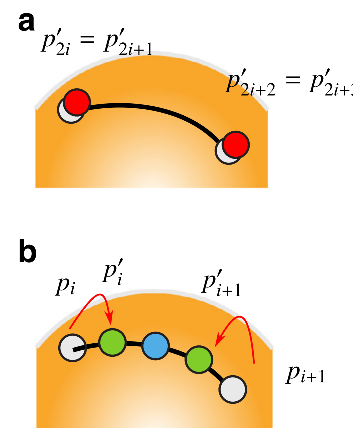

Fig. 1. Illustration of the dual subdivision operations. (a) Initial point duplication step I_D. Newly introduced points are shown in red. (b) Dual averaging step Φ_j, which shrinks edges. New point positions are shown in green, while intermediate points are shown in blue. Image from [2], used with permission. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

The initial subdivision step used to define the dual schemes is a Haar subdivision step (see Stollnitz et al. [16]), denoted by I_D, which duplicates every point $p_i \in \boldsymbol{p}$. That is, the $2n$ points of $I_D(\boldsymbol{p})$ are given by

$$\begin{aligned} I_D(\boldsymbol{p})_{2i} &= p_i, \\ I_D(\boldsymbol{p})_{2i+1} &= p_i. \end{aligned} \tag{1}$$

The dual averaging step is an edge-shrinking operator, denoted by Φ_j, which shrinks every other edge according to the smoothing weight s_j (see Fig. 1). It is given by

$$\Phi_j(\boldsymbol{p})_i = \begin{cases} Interp\left(p_i, p_{i-1}, \frac{s_j}{2}\right) & \text{if } i \bmod 2 = j \bmod 2, \\ Interp\left(p_i, p_{i+1}, \frac{s_j}{2}\right) & \text{otherwise.} \end{cases} \tag{2}$$

Hence, RIA's dual subdivision operation is given by

$$\Phi_{m-1} \cdots \Phi_1 \cdot \Phi_0 \cdot I_D(\boldsymbol{p}).$$

The initial subdivision step used to define the primal schemes is a linear subdivision step, denoted by I_P, which introduces a midpoint between every pair of points in $\boldsymbol{p}$. The $2n$ points of $I_P(\boldsymbol{p})$ are given by

$$\begin{aligned} I_P(\boldsymbol{p})_{2i} &= p_i, \\ I_P(\boldsymbol{p})_{2i+1} &= Interp\left(p_i, p_{i+1}, \frac{1}{2}\right). \end{aligned} \tag{3}$$

The primal averaging step is a modified Laplacian operator, denoted by Λ_j, which moves every second point towards the midpoint of its neighbours according to the smoothing weight s_j (see Fig. 2). It is given by

$$\Lambda_j(\boldsymbol{p})_i = \begin{cases} p_i & \text{if } i \bmod 2 = j \bmod 2, \\ Interp(p_i, m_i, s_j) & \text{otherwise,} \end{cases} \tag{4}$$

where $m_i = Interp(p_{i-1}, p_{i+1}, \frac{1}{2})$. Hence, RIA's primal subdivision operation is given by

$$\Lambda_{m-1} \cdots \Lambda_1 \cdot \Lambda_0 \cdot I_P(\boldsymbol{p}).$$

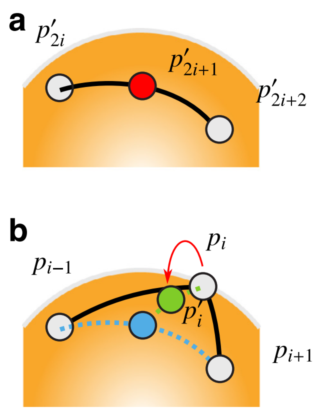

Fig. 2. Illustration of the primal subdivision operations. (a) Initial linear subdivision step I_P. Newly introduced points are shown in red. (b) Primal averaging step Λ_j, which is a modified Laplacian. New point positions are shown in green, while intermediate points are shown in blue. Image from [2], used with permission. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

By construction, both Φ_j and Λ_j are locally invertible[2], and can be inverted by replacing s_j with $\frac{s_j}{s_j - 1}$ in Eqs. (2) and (4).

Reverse subdivision via RIA is accomplished by applying the inverse averaging steps in reverse order, followed by a basic reverse of the initial subdivision. The simplest reverses for I_D and I_P produce n points from $2n$ points, and are described by

$$\hat{I}_D(\boldsymbol{p})_i = Interp\left(p_{2i}, p_{2i+1}, \frac{1}{2}\right), \tag{5}$$

for the dual reverse, and

$$\hat{I}_P(\boldsymbol{p})_i = p_{2i}, \tag{6}$$

for the primal reverse. Hence, RIA's dual and primal reverse subdivision operations are respectively given by

$$\hat{I}_D \cdot \Phi_0^{-1} \cdot \Phi_1^{-1} \cdots \Phi_{m-1}^{-1}(\boldsymbol{p})$$

and

$$\hat{I}_P \cdot \Lambda_0^{-1} \cdot \Lambda_1^{-1} \cdots \Lambda_{m-1}^{-1}(\boldsymbol{p}).$$

In order to perfectly reconstruct the original curve from a reverse subdivided/coarse curve, some multiresolution details must be calculated prior to application of $\hat{I}_D$ or $\hat{I}_P$. Let $\boldsymbol{q} = \Phi_0^{-1} \cdot \Phi_1^{-1} \cdots \Phi_{m-1}^{-1}(\boldsymbol{p})$ in the dual case or $\boldsymbol{q} = \Lambda_0^{-1} \cdot \Lambda_1^{-1} \cdots \Lambda_{m-1}^{-1}(\boldsymbol{p})$ in the primal case. The details encapsulate the differences between $\boldsymbol{q}$ and the expected output of subdividing its coarse version, e.g., $I_D \circ \hat{I}_D(\boldsymbol{q})$. In the dual case, these details are

$$\vec{d_i} = \text{ half the difference between } q_{2i} \text{ and } q_{2i+1}, \tag{7}$$

and in the primal case, they are

$$\vec{d_i} = \text{ the difference between } q_{2i+1} \text{ and } Interp\left(q_{2i}, q_{2i+2}, \frac{1}{2}\right). \tag{8}$$

These details can be represented using vectors in Euclidean space or rotations in spherical space, and can be found as the solution of an inverse geodesic problem. After the coarse points are subdivided using I_D or I_P, they are restored using the direct geodesic problem.[3]

[2] Local invertibility here refers to the ability to recompute each original point position based on a local neighbourhood of points. It allows the inverses to be defined in terms of *Interp* operations, and was produced by separate treatments for every other edge/point and the restriction that $s_j \neq 1$.

[3] The existence of an interpolation method, *Interp*, implies that the direct and inverse geodesic problems can be solved.

As an additional note, these details can also be used to adjust the basic reverse schemes as desired. In order to improve the behaviour of $\hat{I}_P$, e.g., we can add $\vec{d}_{i-1}$ and $\vec{d_i}$ (scaled by a factor of $\frac{1}{3}$) to $\hat{I}_P(\boldsymbol{q})_i$ for each i. These are then subtracted out before applying the forward subdivision I_P.

4. Review of NURBS

NURBS curves can be defined and understood in terms of a geometric interpretation we refer to as the “lift-project” approach. Given points $\boldsymbol{p} = [p_0, p_1, \ldots, p_{n-1}]^T$, $p_i \in \mathbb{R}^3$, with corresponding weights $\boldsymbol{w} = [w_0, w_1, \ldots, w_{n-1}]^T$, $w_i > 0$, a NURBS curve of order k is defined parametrically as follows:

$$R(u) = \frac{\sum_{i=0}^{n-1} N_{i,k}(u) \cdot w_i \cdot p_i}{\sum_{i=0}^{n-1} N_{i,k}(u) \cdot w_i}, \tag{9}$$

where $N_{i,k}(u)$ is the ith B-Spline basis function of order k.

Multiplication of p_i by w_i *lifts* the point p_i out of affine space (the space of valid points) into projective space. The numerator of Eq. (9) defines a B-Spline curve between the lifted points $w_i p_i$ in projective space. Division by the denominator *projects* the resulting curve points back into affine space, producing a valid NURBS curve.

This methodology can be additionally employed in the service of NURBS subdivision [17]. Here, the points p_i are lifted, the lifted points $w_i p_i$ are subdivided in projective space, and then the subdivided points are projected into affine space. Typically, the points $w_i p_i$ are represented using homogeneous coordinates, where the extra coordinate holds the weights of the associated points/the factors of projection. It can be observed that the values of the extra coordinate for the subdivided points comes from subdividing the weights w_i.

For curves in Euclidean space, the use of the lift-project method together with RIA is clear. Consider (without loss of generality) the primal case. Let L_{-1} denote the lifting operation, represented as a scaling matrix formed by placing the weights w_i along the diagonal. Let L_m^{-1} denote the projection operation, formed by placing the factors for projection (the subdivided weights, inverted) along the diagonal. These are denoted by

$$L_{-1} = diag(\boldsymbol{w}), \quad L_m^{-1} = diag(\Lambda_{m-1} \cdots \Lambda_1 \cdot \Lambda_0 \cdot I_P(\boldsymbol{w}))^{-1}.$$

The subdivision operation thus becomes

$$L_m^{-1} \cdot \Lambda_{m-1} \cdots \Lambda_1 \cdot \Lambda_0 \cdot I_P \cdot L_{-1}(\boldsymbol{p}),$$

and the reverse is

$$L_{-1}^{-1} \cdot \hat{I}_P \cdot \Lambda_0^{-1} \cdot \Lambda_1^{-1} \cdots \Lambda_{m-1}^{-1} \cdot L_m(\boldsymbol{p}).$$

However, this approach does not generalize easily to non-Euclidean spaces. The behaviours of L_{-1} and L_m^{-1} are not well-understood outside of Euclidean space and, even if they were, the interpolation operation *Interp* is not necessarily defined in the associated projective space. A consequence of this is that it makes little sense to use homogeneous coordinates to represent the points, and so we must handle the p_i and w_i separately.

5. RIA for NURBS

As its construction in terms of *Interp* operations (such as *SLERP*) is the key aspect of RIA that allows it to be generalized to the sphere and other manifolds, it is essential that the points $\boldsymbol{p}$ not be lifted outside of the space in which *Interp* is defined (e.g. the surface of the sphere). Therefore, the effects of the lift-project method must be somehow incorporated into individual *Interp* operations.

This can be achieved by incorporating the vertex weights w_i into the interpolation parameter u of *Interp*(p, q, u). Whereas the u values of the unmodified RIA framework are functions of the s_j

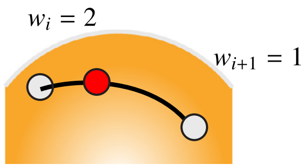

Fig. 3. Illustration of the initial NURBS subdivision in the primal case, with example weights. The position of the subdivided point is determined by the new interpolation parameter, $u = \frac{w_{i+1}}{w_i + w_{i+1}} = \frac{1}{3}$.

(see, e.g., Eqs. (2) and (4)), the u values of the modified framework are functions of both the s_j and w_i.

Consider, for instance, a primal subdivision in Euclidean space that consists of no averaging steps (i.e. only the basic subdivision I_P with $Interp = LERP$). We store the 3D points $p_i \in \mathbb{R}^3$ and the vertex weights $w_i > 0 \in \mathbb{R}$ in separate vectors $\boldsymbol{p}$ and $\boldsymbol{w}$. The NURBS subdivision for this situation is given by $\mathcal{I}_P(\boldsymbol{p}) = L_0^{-1} \cdot I_P \cdot L_{-1}(\boldsymbol{p})$, and the vertex weights of the subdivided points are given by $I_P(\boldsymbol{w})$.

In this situation, it can be observed that the structure of $\mathcal{I}_P$ is very similar to that of I_P:

$$\begin{bmatrix} \frac{1}{w_0} & & & \\ & \frac{2}{w_0+w_1} & & \\ & & \frac{1}{w_1} & \\ & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ \frac{1}{2} & \frac{1}{2} & \\ & 1 & \\ & & \ddots \end{bmatrix} \cdot \begin{bmatrix} w_0 & & \\ & w_1 & \\ & & \ddots \end{bmatrix} = \begin{bmatrix} 1 & & \\ \frac{w_0}{w_0+w_1} & \frac{w_1}{w_0+w_1} & \\ & 1 & \\ & & \ddots \end{bmatrix}.$$

Whereas I_P introduces a midpoint between p_0 and p_1, $\mathcal{I}_P$ introduces a point between p_0 and p_1 whose position is varied according to the weights w_0 and w_1. In the generalized setting, the points $\mathcal{I}_P(\boldsymbol{p})$ are given by

$$\begin{aligned} \mathcal{I}_P(\boldsymbol{p})_{2i} &= p_i, \\ \mathcal{I}_P(\boldsymbol{p})_{2i+1} &= Interp\Big(p_i,\, p_{i+1},\, \tfrac{w_{i+1}}{w_i+w_{i+1}}\Big), \end{aligned} \tag{10}$$

while the weights remain as $I_P(\boldsymbol{w})$ (with $Interp = LERP$). See Fig. 3 for an illustration of $\mathcal{I}_P$ in spherical space.

Now, if we consider a primal subdivision in Euclidean space with one averaging step, we obtain a scheme of the form $L_1^{-1} \cdot \Lambda_0 \cdot I_P \cdot L_{-1}(\boldsymbol{p})$. Once we lift $\boldsymbol{p}$ out of affine space into projective space using L_{-1}, the operations become non-affine and cannot be formulated as sequences of $Interp$ without performing a corresponding projection (in this case, L_1^{-1}). If only a single projection L_m^{-1} is used to return to affine space, then the modular design of RIA based on sequences of simple operations breaks down.

Hence, the core idea behind this work is to utilize several projection and lifting operations throughout the subdivision process, allowing individual operations to be made affine and formulated using $Interp$. For instance, we have already seen that projecting by $L_0^{-1} = diag(I_P(\boldsymbol{w}))^{-1}$ allows us to return to affine space after applying L_{-1} and I_P in sequence, creating an initial NURBS subdivision $\mathcal{I}_P$. Immediately afterwards, we can lift by L_0 in order to return to projective space, where we average by Λ_0 and return to affine space with L_1^{-1}. This produces an affine NURBS averaging step, $\boldsymbol{\Lambda}_0 = L_1^{-1} \cdot \Lambda_0 \cdot L_0$.

Formally,

$$\begin{aligned} & L_1^{-1} \cdot \Lambda_0 \cdot I_P \cdot L_{-1}(\boldsymbol{p}) \\ &= L_1^{-1} \cdot \Lambda_0 \cdot \big(L_0 \cdot L_0^{-1}\big) \cdot I_P \cdot L_{-1}(\boldsymbol{p}) \\ &= \big(L_1^{-1} \cdot \Lambda_0 \cdot L_0\big) \cdot \big(L_0^{-1} \cdot I_P \cdot L_{-1}\big)(\boldsymbol{p}) \\ &= \boldsymbol{\Lambda}_0 \cdot \mathcal{I}_P(\boldsymbol{p}). \end{aligned}$$

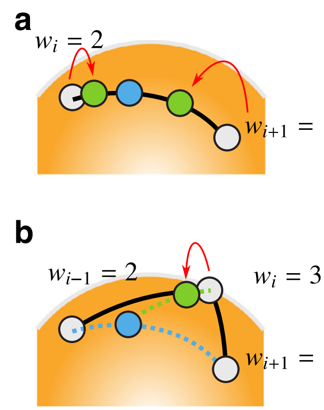

Fig. 4. Illustration of the NURBS averaging operations, with example vertex weights. (a) Dual averaging step $\boldsymbol{\Phi}_j$. (b) Primal averaging step $\boldsymbol{\Lambda}_j$. The vertex weights affect the interpolation method parameters only, causing new point positions to be pulled or pushed along geodesic lines.

This approach can be extended to arbitrarily many averaging steps. Let us define $\boldsymbol{w}^{[i]} = [w_0^{[i]}, w_1^{[i]}, \ldots, w_{2n-1}^{[i]}]^T$, the vector of weights after the initial subdivision and i averaging steps, and L_i, the diagonalization of $\boldsymbol{w}^{[i]}$:

$$\begin{aligned} \boldsymbol{w}^{[i]} &= \Phi_{i-1} \cdots \Phi_1 \cdot \Phi_0 \cdot I_D(\boldsymbol{w}) \\ &\text{or} \qquad\qquad L_i = diag(\boldsymbol{w}^{[i]}). \\ \boldsymbol{w}^{[i]} &= \Lambda_{i-1} \cdots \Lambda_1 \cdot \Lambda_0 \cdot I_P(\boldsymbol{w}) \end{aligned}$$

(Note that the weights are always processed in Euclidean fashion, i.e., with $Interp = LERP$.)

In the dual case, the initial subdivision I_D is replaced by $\mathcal{I}_D = L_0^{-1} \cdot I_D \cdot L_{-1} = I_D$, whose action on $\boldsymbol{p}$ is given by Eq. (1). The averaging steps Φ_j are replaced by $\boldsymbol{\Phi}_j = L_{j+1}^{-1} \cdot \Phi_j \cdot L_j$, given by

$$\boldsymbol{\Phi}_j(\boldsymbol{p})_i = \begin{cases} Interp\Big(p_i,\, p_{i-1},\, \frac{s_j w_{i-1}^{[j]}}{2w_i^{[j+1]}}\Big) & \text{if } i \bmod 2 = j \bmod 2, \\ Interp\Big(p_i,\, p_{i+1},\, \frac{s_j w_{i+1}^{[j]}}{2w_i^{[j+1]}}\Big) & \text{otherwise.} \end{cases} \tag{11}$$

See Fig. 4(a) for an illustration. Intuitively, while Φ_j will shrink every other edge towards its midpoint, $\boldsymbol{\Phi}_j$ pulls each midpoint towards the endpoint with greatest weight and shrinks towards that.

The dual scheme now becomes $\boldsymbol{\Phi}_{m-1} \cdots \boldsymbol{\Phi}_1 \cdot \boldsymbol{\Phi}_0 \cdot \mathcal{I}_D(\boldsymbol{p})$, or equivalently

$$\big(L_m^{-1} \cdot \Phi_{m-1} \cdot L_{m-1}\big) \cdots \big(L_1^{-1} \cdot \Phi_0 \cdot L_0\big)\big(L_0^{-1} \cdot I_D \cdot L_{-1}\big)(\boldsymbol{p}),$$

and the subdivided points have weights $\boldsymbol{w}^{[m]}$.

In the primal case, the initial subdivision I_P is replaced by $\mathcal{I}_P = L_0^{-1} \cdot I_P \cdot L_{-1}$, whose action on $\boldsymbol{p}$ is given by Eq. (10). The averaging steps Λ_j are replaced by $\boldsymbol{\Lambda}_j = L_{j+1}^{-1} \cdot \Lambda_j \cdot L_j$, given by

$$\boldsymbol{\Lambda}_j(\boldsymbol{p})_i = \begin{cases} p_i & \text{if } i \bmod 2 = j \bmod 2, \\ Interp\Big(p_i,\, m_i',\, \frac{s_j (w_{i-1}^{[j]} + w_{i+1}^{[j]})}{2w_i^{[j+1]}}\Big) & \text{otherwise,} \end{cases} \tag{12}$$

where $m_i' = Interp\Big(p_{i-1},\, p_{i+1},\, \frac{w_{i+1}^{[j]}}{w_{i-1}^{[j]} + w_{i+1}^{[j]}}\Big)$. See Fig. 4(a) for an illustration. Intuitively, while Λ_j moves every second point towards the Laplacian (the midpoint of its neighbours), $\boldsymbol{\Lambda}_j$ pulls the Laplacian towards the neighbour with greater weight. How strongly the point is then moved to this pulled Laplacian depends on its weight versus the weights of its neighbours.

The primal scheme now becomes $\mathbf{\Lambda}_{m-1}\cdots\mathbf{\Lambda}_1\cdot\mathbf{\Lambda}_0\cdot\mathcal{I}_P(\boldsymbol{p})$, or equivalently

$$\left(L_m^{-1}\cdot\Lambda_{m-1}\cdot L_{m-1}\right)\cdots\left(L_1^{-1}\cdot\Lambda_0\cdot L_0\right)\left(L_0^{-1}\cdot I_P\cdot L_{-1}\right)(\boldsymbol{p}),$$

and the subdivided points have weights $\boldsymbol{w}^{[m]}$.

As before, the averaging steps $\mathbf{\Phi}_j$ and $\mathbf{\Lambda}_j$ can be inverted by replacing s_j with $\frac{s_j}{s_j-1}$ in Eqs. (11) and (12). These inverses are applied in reverse order prior to a basic scheme that reverses the initial subdivision.

The basic reverse schemes $\hat{I}_D$ and $\hat{I}_P$ can be kept in their original forms (see Eqs. (5) and (6)), though multiresolution details must be calculated differently in the primal case. These new details are

$$\vec{d_i} = \text{ the difference between } q_{2i+1} \text{ and } Interp\left(q_{2i}, q_{2i+2}, \frac{w^{[0]}_{2i+2}}{w^{[0]}_{2i}+w^{[0]}_{2i+2}}\right). \tag{13}$$

where $\boldsymbol{q}=\mathbf{\Lambda}_0^{-1}\cdot\mathbf{\Lambda}_1^{-1}\cdots\mathbf{\Lambda}_{m-1}^{-1}(\boldsymbol{p})$. Note that for NURBS curves, details also need to be calculated for the weights as well. These details are calculated using Eqs. (7) and (8), with $w_i^{[0]}$ in place of q_i and $Interp = LERP$.

6. Weights for geospatial vectors

One of the primary motivations behind RIA and this work lies in the multiscale representation of geospatial vector data. As geospatial sensors continue to improve, geospatial vector data are gathered at increasingly high resolutions, necessitating the development of methods to deal with their incredible size [18].

Multiresolution frameworks allow such big data to be decomposed into a coarse approximation, along with details needed to losslessly restore the original data. Potential applications of these frameworks include data compression, data transmission, level-of-detail rendering, and improved query times for vector data – all of which benefit when the coarse approximations more closely resemble the original data.

As multiresolution details $\vec{d_i}$ encapsulate the differences between the coarse approximation and the original vector, we can improve the quality of the coarse approximation by minimizing the magnitudes of these details. In the primal case, by inspection of Eq. (13) it can be observed that this occurs when the detail $\vec{d_i}$ is as perpendicular as possible to the great circle arc between q_{2i} and q_{2i+2}, and can be manipulated by adjusting the vertex weights w_i (see Fig. 5).

Hence, we have developed an optimization process that automatically assigns vertex weights w_i to the p_i, for the purposes of improving the coarse approximations. Note that we only cover primal schemes on the sphere.

Given a vector of points $\boldsymbol{p}$, we first determine $\boldsymbol{q}=\mathbf{\Lambda}_0^{-1}\cdot\mathbf{\Lambda}_1^{-1}\cdots\mathbf{\Lambda}_{m-1}^{-1}(\boldsymbol{p})$. Then, for each i, we project q_{2i+1} onto the great circle passing through q_{2i} and q_{2i+2}, giving an optimal pulled Laplacian m_i, and determine the value u_i such that $m_i = Interp(q_{2i}, q_{2i+2}, u_i)$. We clamp u_i to the range $[\frac{1}{4},\frac{3}{4}]$ in order to avoid the creation of excessively large w_i.

These u_i can now be used to constrain the $w_{2i}^{[0]}$ via the relation (refer to Eq. (13))

$$u_i = \frac{w^{[0]}_{2i+2}}{w^{[0]}_{2i}+w^{[0]}_{2i+2}}$$
$$u_i\cdot(w^{[0]}_{2i}+w^{[0]}_{2i+2}) = w^{[0]}_{2i+2}$$
$$u_i\cdot w^{[0]}_{2i}+(u_i-1)\cdot w^{[0]}_{2i+2}=0,$$

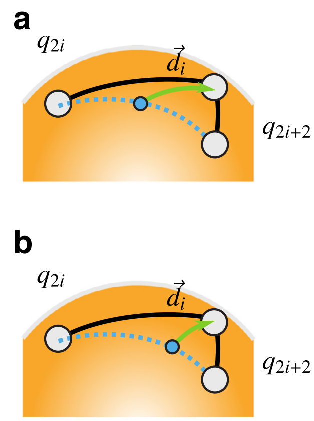

Fig. 5. The magnitude of a detail $\vec{d_i}$ can be reduced by properly adjusting vertex weights. (a) The points q_{2i} and q_{2i+2} have equal weights. (b) If the weight of q_{2i+2} is increased relative to that of q_{2i}, the intermediate point (in blue) is pulled and $\vec{d_i}$ becomes smaller. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

which gives rise to the following linear system:

$$\begin{bmatrix} u_0 & u_0-1 & & & \\ & u_1 & & u_1-1 & \\ & & & u_2 & \\ & & & & \ddots \end{bmatrix}\begin{bmatrix} w_0^{[0]} \\ w_2^{[0]} \\ w_4^{[0]} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

In order to avoid the trivial solution $w_{2i}^{[0]}=0$, we softly constrain $w_{2i}^{[0]}\approx 1$:

$$\left[\begin{array}{ccccc} 1 & & & & \\ & 1 & & & \\ & & & 1 & \\ & & & & \ddots \\ \hline u_0 & u_0-1 & & & \\ & u_1 & & u_1-1 & \\ & & & u_2 & \\ & & & & \ddots \end{array}\right]\begin{bmatrix} w_0^{[0]} \\ w_2^{[0]} \\ w_4^{[0]} \\ \vdots \end{bmatrix} = \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ \vdots \\ \hline 0 \\ 0 \\ 0 \\ \vdots \end{array}\right].$$

Solving this linear system provides optimized weights $w_{2i}^{[0]}$. The odd numbered weights can be inferred by, e.g., taking the midpoints $w_{2i+1}^{[0]}=\frac{1}{2}w_{2i}^{[0]}+\frac{1}{2}w_{2i+2}^{[0]}$. This produces a complete $\boldsymbol{w}^{[0]}$, from which a set of weights $\boldsymbol{w}$ can be obtained via $\boldsymbol{w}=\Lambda_{i-1}\cdots\Lambda_1\cdot\Lambda_0(\boldsymbol{w}^{[0]})$.

Note that if $\boldsymbol{p}$ is not the high resolution version of the data (i.e., it has been reverse subdivided), then the high resolution data will be altered by the assignment of new vertex weights. In order to preserve the original data while using the new weights, the new weights should be subdivided to the original resolution (pre-existing weight details may be used), and then assigned to the original set of points.

7. Results

In this section, we present some result figures generated using the modified RIA framework. Figs. 6–8 were created using a degree 3 subdivision, via the primal scheme using smoothing weights $S=\{\frac{1}{2}\}$. Three applications of this subdivision were used to produce the high resolution version of each curve. Reverse schemes use the improved $\hat{I}_P$ described at the end of Section 3.

Fig. 6 illustrates an example of multiscale editing on a spherical NURBS curve. The letter '*S*' has been modeled on the sphere, with vertex weights increased at the ends of the '*S*' shape. Edits

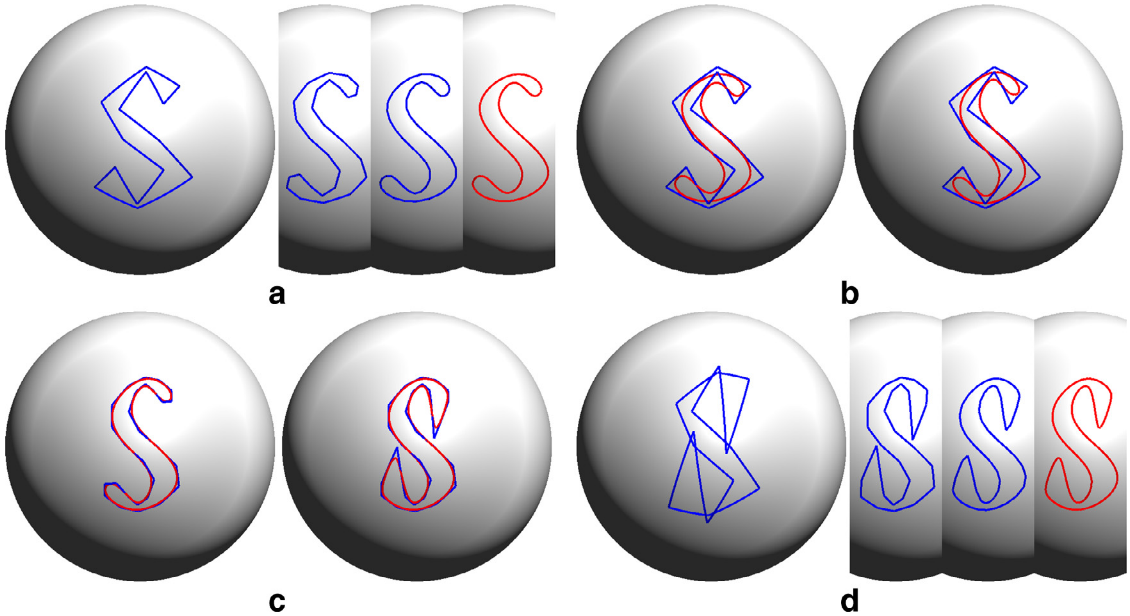

Fig. 6. Multiscale editing of the letter '*S*'. (a) Control polygon and subdivided versions. (b) The bottom and top of the '*S*' are adjusted by changing vertex weights. (c) The ends of the '*S*' are adjusted by moving vertices and changing weights. (d) Edited control polygon and subdivided versions, after optimizing weights at the highest resolution.

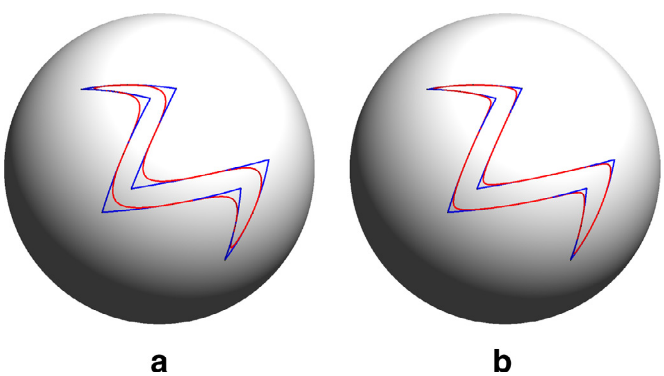

Fig. 7. By adjusting vertex weights in the control polygon (shown in blue), the shape of the subdivided curve (shown in red) can be finely controlled. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

can be applied on both the subdivided and reverse subdivided versions without loss of detail.

Fig. 7 highlights the impact of increased vertex weights on a subdivided spherical curve. By increasing a point's weight, the curve is pulled more strongly towards that point. Thus, different spherical curves can be produced without changing the control polygon.

An example of subdivision on the surface of an ellipsoid is shown in Fig. 8. Both B-Spline curves and NURBS curves may be represented in a multiscale manner on the ellipsoid (here an ellipsoid with a flattening factor of 0.5) using the modified RIA framework with an appropriate interpolation method. Our interpolation method was constructed by combining open-source solutions to the direct and inverse geodesic problems for ellipsoids (see [9,19]).

In Fig. 9, we illustrate an application of this framework to the multiscale representation of geospatial data. The border of Mexico has been reverse subdivided four times using a primal scheme with $S = \{\frac{1}{4}\}$. Optimized weight values were computed at each level using the method from Section 6. The resulting weights ranged from 0.65 to 1.11, with an average of 0.888 and standard deviation of 0.108 (see Fig. 9(c) for a histogram).

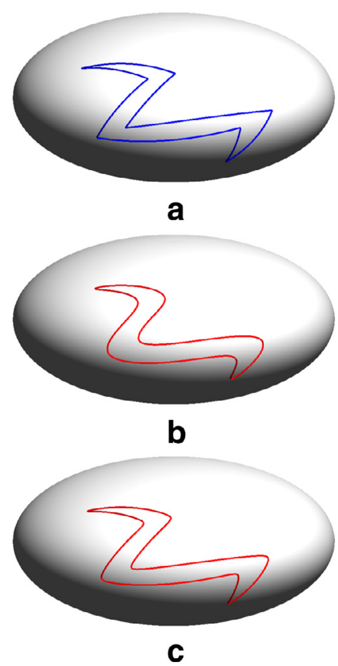

Fig. 8. Subdivision on an ellipsoid of revolution. (a) Coarse control polygon. (b) Subdivided B-Spline curve. (c) Subdivided NURBS curve.

We have compared the accuracy of our geospatial results with those from two other methods, namely subsampling ($\hat{I}_P$ from Eq. (6)) and the unmodified RIA framework (primal, with $S = \{\frac{1}{4}\}$). The results of this comparison are contained in Table 1. For $n = 1, 2, 3, 4,$ we took the Mexican border, reverse subdivided n times, discarded the details, subdivided n times, and then determined the errors/distances between the vertices of the resulting subdivided border and the original border (on the unit sphere).

On average, our modified framework with optimized weights can more closely approximate the original curve when details are missing, and with less variance overall. As a result, we suspect that this framework may prove useful for applications such as compression, data transmission, etc.

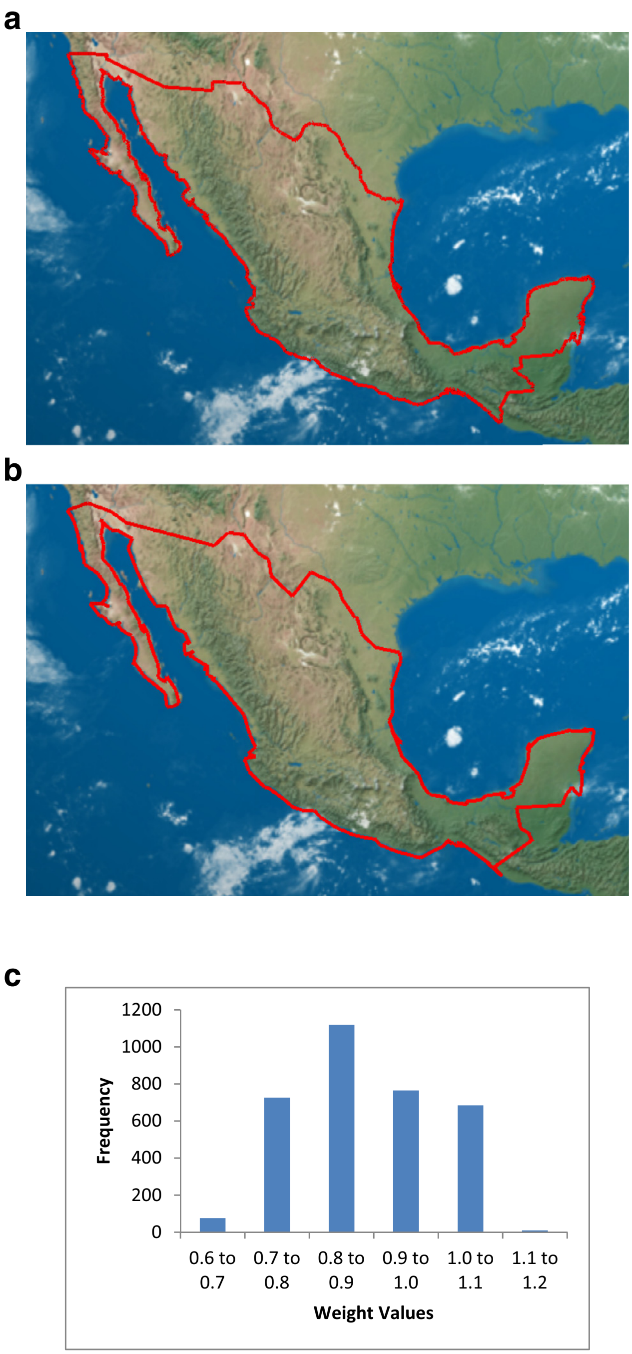

Fig. 9. Reverse subdivision on geospatial data. (a) High resolution border of Mexico. (b) The border of Mexico after four reverse subdivisions (primal scheme with $S = \{\frac{1}{4}\}$ and optimized vertex weights). (c) Histogram of the optimized weights. Texture image for the Earth courtesy of www.shadedrelief.com.

Table 1
Average error μ and standard deviation σ ($\times 10^{-3}$) between the border of Mexico and its subdivided approximations.

Number of Reverses	Subsampling	RIA with $S = \{\frac{1}{4}\}$	RIA for NURBS with $S = \{\frac{1}{4}\}$
1	$\mu = 0.166$	$\mu = 0.201$	$\mu = 0.169$
	$\sigma = 0.361$	$\sigma = 0.271$	$\sigma = 0.228$
2	$\mu = 0.406$	$\mu = 0.404$	$\mu = 0.359$
	$\sigma = 0.604$	$\sigma = 0.441$	$\sigma = 0.397$
3	$\mu = 0.828$	$\mu = 0.762$	$\mu = 0.694$
	$\sigma = 0.935$	$\sigma = 0.735$	$\sigma = 0.657$
4	$\mu = 1.589$	$\mu = 1.471$	$\mu = 1.354$
	$\sigma = 1.645$	$\sigma = 1.343$	$\sigma = 1.173$

8. Conclusions

In this work, we have presented a modified repeated invertible averaging framework that allows one to subdivide or reverse subdivide the control points of a spherical or ellipsoidal NURBS curve. Using an appropriate interpolation method, this framework can be utilized in other spaces and on other manifolds. Our modification circumvents issues with the lift-project method by incorporating its effect directly into the weight parameter of the chosen interpolation method. We have also provided a method to automatically assign weights to the vertices of a given vector, and illustrated this framework's usefulness in a geospatial context.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Acknowledgments

Funding for this research was provided by an NSERC CRD with our industrial collaborator, Global Grid Systems (formerly the PYXIS innovation).

References

[1] Piegl L. On NURBS: A survey. IEEE Comput Gr Appl 1991;11(1):55–71.
[2] Alderson TF, Mahdavi-Amiri A, Samavati FF. Multiresolution on spherical curves. Gr Model 2016;86:13–24.
[3] Shoemake K. Animating rotation with quaternion curves. In: Proceedings of SIGGRAPH 1985; 1985. p. 245–54.
[4] Huang K-T, Gong H, Fang FZ, Li ZJ. Generation of spherical non-uniform rational basis spline curves and its application in five-axis machining. Proc Inst Mech Eng B: J Eng Manuf 2015;229(7):1104–17.
[5] Kaplan CS, Salesin DH. Islamic star patterns in absolute geometry. ACM Trans Gr 2004;23(2):97–119.
[6] Hamekasi N, Samavati F. Designing Persian floral patterns using circle packing. In: Proceedings of the international conference on computer graphics theory and applications; 2012. p. 135–42.
[7] Lane JM, Riesenfeld RF. A theoretical development for the computer generation and display of piecewise polynomial surfaces. IEEE Trans Pattern Anal Mach Intell 1980;2(1):35–46.
[8] do Carmo MP. Differential geometry of curves and surfaces, chap. 4-6. Prentice-Hall, Inc.; 1976.
[9] Karney CFF. Algorithms for geodesics. J Geodesy 2013;87(1):43–55.
[10] Buss SR, Fillmore JP. Spherical averages and applications to spherical splines and interpolation. ACM Trans Gr 2001;20(2):95–126.
[11] Schaefer S, Goldman R. Freeform curves on spheres of arbitrary dimension. In: Proceedings of Pacific graphics 2005; 2005. p. 160–2.
[12] Bartels RH, Samavati FF. Reversing subdivision rules: Local linear conditions and observations on inner products. J Comput Appl Math 2000;119(1–2):29–67.
[13] Samavati FF, Bartels RH, Olsen L. Local B-spline multiresolution with examples in iris synthesis and volumetric rendering. In: Image pattern recognition: synthesis and analysis in biometrics. In: Series in machine perception and artificial intelligence, 67. World Scientific Publishing; 2007. p. 65–102.
[14] Bessel FW. Über die Berechnung der geo-graphischen längen und breiten aus geodätischen vermessungen [the calculation of longitude and latitude from geodesic measurements]. Astron Nachr 1825;331(8):852–61. (Translated by C.F.F. Karney and R.E. Deakin, 2010).
[15] Vincenty T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Surv Rev 1975;23(176):88–93.
[16] Stollnitz EJ, Derose TD, Salesin DH. Wavelets for computer graphics: theory and applications. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1996. ISBN 1-55860-375-1.
[17] Piegl L, Tiller W. 2nd edition. Springer-Verlag Berlin Heidelberg; 1997.
[18] Mahdavi-Amiri A, Alderson T, Samavati F. A survey of digital Earth. Comput Gr 2015;53, Part B:95–117.
[19] Karney C. GeographicLib. 2019. [Online; accessed 08-March-2019] https://geographiclib.sourceforge.io/.