Smoothly Deformable Spheres: Modeling, Deformation, and Interaction

Daniel Schmitter* EPFL
Pablo García-Amorena† EPFL
Michael Unser‡ EPFL

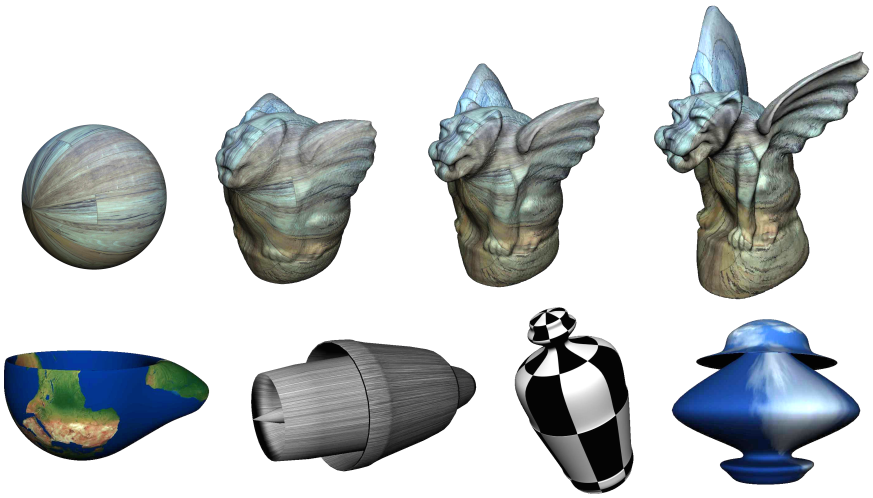

Figure 1: *Smooth modeling of shapes with spherical toplogy. The continuous deformation of the sphere into the Gargoyle is shown in the top row where a wood texture has been added to the surface. The shapes in the bottom row consist of a single surface patch and are constructed through the interactive deformation of the sphere. The interpolating structure of the model allows us to intuitively design surfaces that can adopt shapes beyond the classical spherical topology. Our framework is inherently smooth, which facilitates natural texturing.*

Abstract

Existing shape models with spherical topology are typically designed either in the *discrete* domain using *interpolating* polygon meshes or in the continuous domain using *smooth* but *non-interpolating* schemes such as NURBS. Polygon models and subdivision methods require a large number of parameters to model smooth surfaces. NURBS need fewer parameters but have a complicated rational expression and non-uniform shifts in their formulation. We present a new method to construct deformable closed surfaces, which includes the exact sphere, by combining the best of two worlds: a *smooth* and *interpolating* model with a continuously varying tangent plane and well-defined curvature at every point on the surface. Our formulation is simpler than NURBS while it requires fewer parameters than polygon meshes. We demonstrate the generality of our method with applications ranging from intuitive user-interactive shape modeling, continuous surface deformation, reconstruction of shapes from parameterized point clouds, to fast iterative shape optimization for image segmentation.

Keywords: shape modeling, spherical topology, parametric surface, splines, differential geometry

Concepts: •**Computing methodologies** → **Shape modeling;**

*e-mail:daniel.schmitter@epfl.ch
†e-mail:pablo.garcia-amorenagarcia@epfl.ch
‡e-mail:michael.unser@epfl.ch

1 Introduction

The representation of shapes with spherical topology is an ongoing research topic in computer graphics. The reason is the abundant need of closed genus-zero surfaces in industrial, architectural, and animation design as well as biomedical imaging. Designing spherical models that are simultaneously optimal with respect to several different shape characteristics still remains a challenge. Depending on whether an application involves user interaction, shape deformation or optimization schemes, different aspects of a model are more important than others.

On one hand, in user-interactive applications, a fundamental requirement is the possibility to intuitively manipulate the shape. Typically, this presupposes an easy way to directly interact with the surface as well as to control shapes locally. This task is naturally linked to the topic of surface deformation. On the other hand, an application might involve shape deformation as an optimization process. For example, in real-time shape recognition or segmentation, this requires the fast evaluation of derivative- and integral-based quantities in iterative settings. Further, the smoothness of the surface and the number of parameters that are involved can also play an important role. Usually, it is impossible to apply a model that is optimal with respect to all of these requirements. In practice, a compromise is made favoring the most important needs for a specific application. We present a new model to construct deformable shapes with spherical topology that meets all of the above mentioned requirements.

2 Related Work

The most widely used technique to construct deformable spheres in the continuous domain is with NURBS [Piegl and Tiller 2010].

SA '16 Technical Briefs ,, December 05 - 08, 2016, , Macao
DOI: http://dx.doi.org/10.1145/3005358.3005377

NURBS surfaces are based on polynomial B-splines and are defined by a set of control points which allow local shape control. The primary reason for using the rational NURBS expression instead of the non-rational polynomial B-splines is because NURBS are able to exactly reproduce conic sections. Conceptually, this is equivalent to the reproduction of trigonometric functions, which is a necessary requirement for constructing spheres. A drawback of NURBS is their rational form which leads to complicated expressions of related integrals and derivatives [Manni et al. 2011]. Furthermore, the NURBS formulation depends on additional weight parameters which have no intuitive interpretation. Other constructions to approximate sphere-like surfaces based on B-splines have been studied in [Dierckx 1984], whereas in [Delgado-Gonzalo et al. 2013] an exact approach using exponential splines is proposed. Other models use rational Bézier surfaces [Prautzsch et al. 2002], which are also related to splines.

Popular discrete methods are based on polygon meshes [Botsch et al. 2010]. With these models, it is possible to represent shapes of arbitrary topology. Polygon models are interpolating with the control points coinciding with the vertices of the mesh; this implies that the shape is modified by points which directly lie on the boundary of the object. Related to polygon models are subdivision methods used to construct surfaces [Doo and Sabin 1978; DeRose et al. 1998; Catmull and Clark 1998; Stam and Loop 2003]. They are characterized by refinement operations that are iteratively applied to a set of points leading to a continuous limit surface with a certain regularity. One limitation of polygon and subdivision methods is that they require a large number of parameters which can be a challenge when computational speed is required.

3 Spherical Parameterization

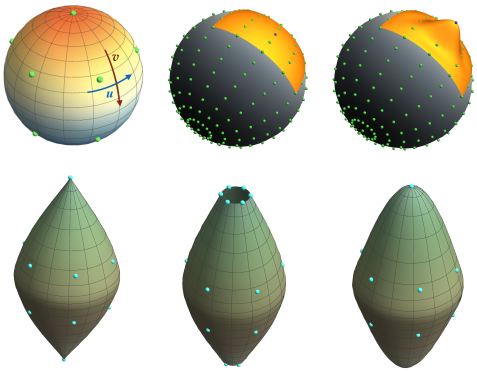

Figure 2: *Top row: Reconstructed sphere with the interpolatory control points shown in light green (left). The yellow region (middle) that is affected by moving a single control point (blue) is shown (right). It corresponds to a patch of size 4×4, which is due to the support of the generator φ_M. Bottom: Closed and smooth deformable sphere. If no smoothness conditions are imposed the surface becomes non-differentiable at the poles (left). If no closeness conditions are imposed the surface looses its spherical topology (middle) when deforming. On the right the closed and deformed sphere is shown with smoothly varying tangent planes at the poles.*

We construct the parametric surface as a tensor product using a weighted sum of integer shifts of a non-rational generator function φ. The surface is expressed as

$$\boldsymbol{\sigma}(u,v) = \sum_{k=0}^{M_1-1} \sum_{l=-1}^{M_2+1} \boldsymbol{c}[k,l]\varphi_{M_1,\mathrm{per}}(u-k)\varphi_{M_2}(v-l),$$

where M_1 and M_2 are the number of control points, $\boldsymbol{c}[k,l] = (c_x[k,l], c_y[k,l], c_z[k,l])$ in the u and v-direction, $\varphi_{M_1,\mathrm{per}} = \sum_{n=-\infty}^{+\infty} \varphi_{M_1}(t - M_1 n)$ is the M_1 periodized basis function and the explicit expression of φ_M is given by (1). The limits in the double sum are due to the compact support of φ_M. From [Schmitter et al. 2015], we know that φ_M exactly reproduces trigonometric functions. Using the property that φ_M also is an interpolator, the exact sphere is parameterized as $\boldsymbol{\sigma}(u,v) = \begin{pmatrix} \cos(2\pi u)\sin(\pi v) \\ \sin(2\pi u)\sin(\pi v) \\ \cos(\pi v) \end{pmatrix}$ with $\boldsymbol{c}[k,l] = \begin{pmatrix} \cos\left(\frac{2\pi k}{M_1}\right)\sin\left(\frac{\pi l}{M_2}\right) \\ \sin\left(\frac{2\pi k}{M_1}\right)\sin\left(\frac{\pi l}{M_2}\right) \\ \cos\left(\frac{\pi l}{M_2}\right) \end{pmatrix}$ and $u, v \in [0,1]$.

Main Result: Smoothness and Closeness Conditions at the Poles. Since $\varphi \in \mathcal{C}^1$, continuity is guaranteed nearly everywhere on the surface as long as the control points do not overlap. However, for the deformed sphere, smoothness is not guaranteed at its poles unless we take appropriate measures. In [Dierckx 1984], it was shown that continuity at the poles is ensured if a deformable sphere is constructed with continuously varying tangent planes at these points. This condition is mathematically expressed as

$$\frac{\partial \boldsymbol{\sigma}}{\partial v}(u,v)\Big|_{v=0} = \boldsymbol{T}_{1,N}\cos(2\pi u) + \boldsymbol{T}_{2,N}\sin(2\pi u), \qquad (2)$$

for the north pole and

$$\frac{\partial \boldsymbol{\sigma}}{\partial v}(u,v)\Big|_{v=1} = \boldsymbol{T}_{1,S}\cos(2\pi u) + \boldsymbol{T}_{2,S}\sin(2\pi u), \qquad (3)$$

for the south pole, where $\boldsymbol{T}_{1,N}$, $\boldsymbol{T}_{2,N}$, $\boldsymbol{T}_{1,S}$, and $\boldsymbol{T}_{2,S}$ are vector parameters that can be freely chosen. Both sides of the equality sign in (2) can independently be simplified; we end up with the condition

$$\boldsymbol{c}[k,-1] = \frac{\boldsymbol{T}_{1,N}\cos\left(\frac{2\pi k}{M_1}\right) + \boldsymbol{T}_{2,N}\sin\left(\frac{2\pi k}{M_1}\right)}{M_2\varphi'_{2M_2}(1)} + \boldsymbol{c}[k,1]. \qquad (4)$$

Similarly, (3) is simplified to

$$\boldsymbol{c}[k,M_2+1] = \boldsymbol{c}[k,M_2-1] - \frac{\boldsymbol{T}_{1,S}\cos\left(\frac{2\pi k}{M_1}\right) + \boldsymbol{T}_{2,S}\sin\left(\frac{2\pi k}{M_1}\right)}{M_2\varphi'_{2M_2}(1)}. \qquad (5)$$

The tangent plane at the poles is then spanned by the vectors $\boldsymbol{T}_{1,N}$, $\boldsymbol{T}_{2,N}$ and $\boldsymbol{T}_{1,S}$, $\boldsymbol{T}_{2,S}$, respectively. Next, we make sure that the sphere remains closed when deforming in order to maintain spherical topology. Again, special attention needs to be paid to the poles; they have to remain being interpolated, *i.e.*, closed. Conceptually, this requires that all the circles of longitude of the original sphere keep originating and ending at the poles of the surface. We denote by $\boldsymbol{c}_N$ and $\boldsymbol{c}_S$ the north and south pole. Now, we can establish the relation

$$\begin{aligned} \boldsymbol{\sigma}(u,0) &= \boldsymbol{c}_N = \boldsymbol{c}[k,0] \quad \text{(north pole)}, \\ \boldsymbol{\sigma}(u,1) &= \boldsymbol{c}_S = \boldsymbol{c}[k,M_2] \quad \text{(south pole)}, \end{aligned}$$

$\forall k \in [0 \ldots M_1 - 1]$. In Figure 2, we illustrate the interpolation and smoothness conditions on the poles.

4 Results and Applications

Intuitive Interactive Modeling. A crucial aspect in interactive shape modeling is that the modeling process must be intuitive. Standard modeling applications allow a user to modify a shape by dragging its control points with the mouse in order to displace them. If

$$\varphi_M(t) = \begin{cases} \frac{\sin^2\left(\frac{\pi}{M}\right)\left(M\sin\left(\frac{2\pi(t-2)}{M}\right)-2\pi(t-2)\right)+\left(M\sin\left(\frac{2\pi}{M}\right)-2\pi\right)\sin^2\left(\frac{\pi(t-2)}{M}\right)}{2\left(\cos\left(\frac{2\pi}{M}\right)-1\right)\left(-M\sin\left(\frac{2\pi}{M}\right)+\pi\cos\left(\frac{2\pi}{M}\right)+\pi\right)} & 1<t<2 \\ \frac{M\left(\sin\left(\frac{2\pi(t-2)}{M}\right)-2\sin\left(\frac{2\pi(t-1)}{M}\right)+\sin\left(\frac{2\pi(t+1)}{M}\right)+\sin\left(\frac{2\pi}{M}\right)-\sin\left(\frac{4\pi}{M}\right)\right)+2\pi t\cos\left(\frac{2\pi}{M}\right)-2\pi(t-1)\cos\left(\frac{4\pi}{M}\right)-2\pi\cos\left(\frac{2\pi t}{M}\right)}{4\left(\cos\left(\frac{2\pi}{M}\right)-1\right)\left(-M\sin\left(\frac{2\pi}{M}\right)+\pi\cos\left(\frac{2\pi}{M}\right)+\pi\right)} & 0<t\leq 1 \\ -\frac{M\left(\sin\left(\frac{2\pi(t-1)}{M}\right)-2\sin\left(\frac{2\pi(t+1)}{M}\right)+\sin\left(\frac{2\pi(t+2)}{M}\right)-\sin\left(\frac{2\pi}{M}\right)+\sin\left(\frac{4\pi}{M}\right)\right)+2\pi\left(t\cos\left(\frac{2\pi}{M}\right)-(t+1)\cos\left(\frac{4\pi}{M}\right)+\cos\left(\frac{2\pi t}{M}\right)\right)}{4\left(\cos\left(\frac{2\pi}{M}\right)-1\right)\left(-M\sin\left(\frac{2\pi}{M}\right)+\pi\cos\left(\frac{2\pi}{M}\right)+\pi\right)} & -1<t\leq 0 \\ \frac{\sin^2\left(\frac{\pi}{M}\right)\left(2\pi(t+2)-M\sin\left(\frac{2\pi(t+2)}{M}\right)\right)+\left(M\sin\left(\frac{2\pi}{M}\right)-2\pi\right)\sin^2\left(\frac{\pi(t+2)}{M}\right)}{2\left(\cos\left(\frac{2\pi}{M}\right)-1\right)\left(-M\sin\left(\frac{2\pi}{M}\right)+\pi\cos\left(\frac{2\pi}{M}\right)+\pi\right)} & -2<t\leq -1 \\ 0 & |t|\geq 2. \end{cases} \tag{1}$$

the control points directly lie on the surface of the shape, the modeling task is significantly simplified. This is the case for polygon models but at the expense of dealing with smooth shapes. On the other hand, NURBS allow for the construction of smooth shapes, but the control points do not interpolate the shape. This makes the modeling task less intuitive. Local shape control is difficult as the surface becomes more complex because it is no longer clear which part of the surface is affected by a specific control point. Our proposed construction solves this problem since φ_M satisfies the interpolation condition and is also smooth. Hence, even if the modeled surface is of great complexity, the modeling process remains intuitive and simple because the control points always lie on the boundary of the shape itself. With relatively few control points, complex structures are easily constructed and modified. Furthermore, due to the compact support of φ_M, local shape control is guaranteed. In Figure 3, we show examples of the use of our framework in an interactive modeling environment. Final renderings, where texture is added to a shape are achieved without discretization artifacts since the underlying nature of the structure is smooth, independently of the number of control points chosen (Figure 1, bottom row).

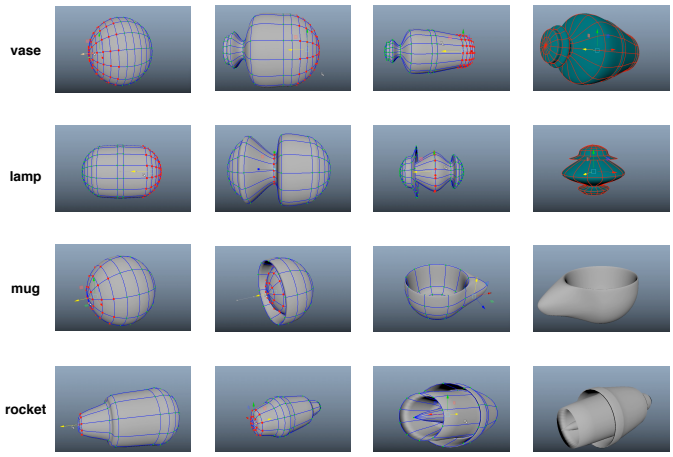

Figure 3: *Implementation of the framework in a shape modeling environment. Different shapes are interactively designed starting from a sphere (from left to right). The interpolatory control points allow to easily model surfaces that can adopt shapes beyond "traditional" spherical topology, such as the mug and rocket.*

Shape Interpolation. When dealing with a parameterized point cloud whose points correspond to the samples of a surface with spherical topology, our formulation allows for an immediate reconstruction of the smooth shape. In this case, for each point $\boldsymbol{p} \in \mathbb{R}^3$ of the point cloud, a pair (u_k, v_l) of coordinates is assigned in the parameter domain and we can establish the relation $\boldsymbol{\sigma}(u_k, v_l) = \boldsymbol{p}_{k,l} = \boldsymbol{c}[k,l]$. For a fixed number of points, M_1, M_2, in the u- and v-direction respectively, the parametric coordinates for the normalized parametric domain, *i.e.*, $u, v \in [0,1]$, are given by $u_k = \frac{k}{M_1}$ and $v_l = \frac{l}{M_2}$. The resulting continuously defined surface $\boldsymbol{\sigma}(u,v)$ is immediately reconstructed since it is fully specified by its control points subject to the smoothness and pole-interpolation conditions described above. An example is shown in Figure 4.

Figure 4: *Interpolation of a parameterized point cloud. The dinosaurs (middle and right) are smooth reconstructions obtained by interpolating the point cloud on the left. Our surface construction is affine-invariant, which implies that a rotation of a shape is simply obtained by rotating the point cloud.*

Smooth Modeling at Arbitrary Resolution. Because our construction of $\boldsymbol{\sigma}$ is inherently smooth the tangent plane and Gaussian curvature are everywhere well-defined even when there are few control points. This can be an advantage to construct textured models with few parameters; for example, in applications involving real-time rendering. As an example, we have parameterized the point cloud of the Gargoyle model using the algorithm described by [Praun and Hoppe 2003]. This allows us to reconstruct a smooth surface by interpolating the points. Additionally, we have subsampled the point cloud at different resolutions to obtain an approximation of the Gargoyle with varying levels of accuracy. In Figure 5, we illustrate the result and show a comparison with polygons.

Efficient Shape Deformation. An advantage of our continuous-domain model is that the shapes are described by a finite number of control points, whereas the corresponding coordinate functions x, y, and z live in a continuum. This allows us to describe a shape deformation process in the continuous domain by just displacing the control points. An example of a deformation is shown in Figure 1.

Fast Computation of Surface Integrals. In certain applications that require iterative optimization, it is necessary to compute surface or volume integrals in a fast way. An example is the deformation of a surface which is guided by minimizing an energy functional in real time. We illustrate how a flux E across the surface S, parameterized by $\boldsymbol{\sigma}(u,v)$, is computed in an efficient and fast way. Given a vector field $\boldsymbol{f}$, one way of expressing the flux E is by

$$E(\boldsymbol{\sigma}) = \oiint_S \boldsymbol{f} \cdot \mathrm{d}\boldsymbol{S} = \int_0^1 \int_0^1 g^x(\boldsymbol{\sigma}) \mathrm{d}y \wedge \mathrm{d}z, \tag{6}$$

where $\mathrm{d}\boldsymbol{S}$ represents the vector differential of the surface area, $\wedge$ denotes the wedge product, and $g^x(x,y,z) =$

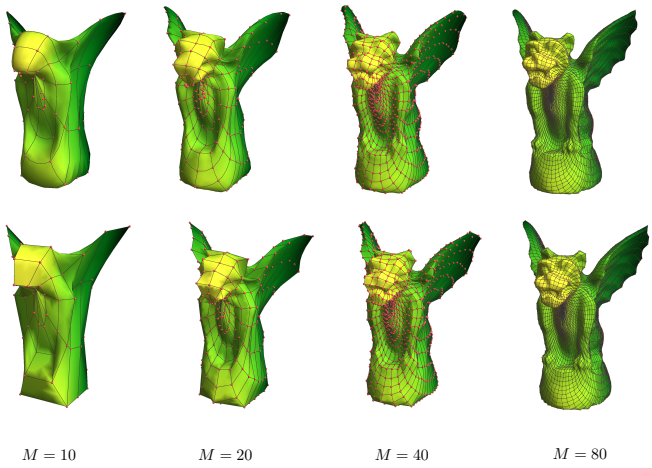

Figure 5: *Interpolation of shapes with spherical topology: Smooth Gargoyle reconstructions at different resolutions. The same number of control points is used in both directions of the parameter domain, i.e., $M = M_1 = M_2$. In the top row, the results obtained with our construction are shown, whereas in the bottom row a polygon reconstruction method is applied. Note that, with our approach the smoothness of the model does not depend on the number of parameters.*

$\int_{-\infty}^{x} \mathrm{div}\boldsymbol{f}(\tau, y, z)\mathrm{d}\tau$ is the pre-integrated divergence of the vector field $\boldsymbol{f}$ along the x-dimension. Typically, $\boldsymbol{f}$ does not depend on the surface and hence, g^x can be precomputed and stored in a look-up table, which significantly speeds up the computation. It is worth mentioning at this point that the use of pre-integrated functions is only possible because we define the surface $\boldsymbol{\sigma}$ in the continuous domain. Next, the flux E can be efficiently optimized by computing the gradient of E w.r.t. the control points using a classical steepest-descent iterative method. An explicit expression of the gradient can easily be obtained and hence, implemented in an exact way. We illustrate the above computation by segmenting the surface of a human brain, which has been extracted from a medical 3D MRI image. We first compute an edgemap of the 3D image using a standard surface extraction algorithm as described by [Aguet et al. 2005] and construct an energy functional E that depends on the gradient of the edgemap. Hence, in (6), the gradient corresponds to $\boldsymbol{f}$. By minimizing (6), $\boldsymbol{\sigma}$ deforms iteratively to approximate the edge map, as shown in Figure 6. The result can easily be manually adjusted by a clinician.

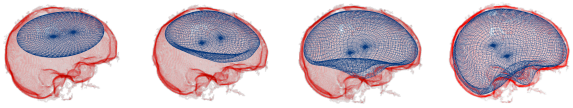

Figure 6: *Brain segmentation in a 3D medical MRI image. The red surface is a rendered edgemap that has been extracted from the medical data. An ellipsoidal surface is initialized inside the brain surface (left) and evolves by iteratively minimizing* (6) *(from left to right). The final result is shown on the right and corresponds to a smooth and continuous closed surface shape.*

5 Conclusion

The standard method for smooth parametric shape modeling in industry is NURBS. In this paper, we presented a framework to model smooth shapes with spherical topology. The fundamental difference with the existing standard is that our basis functions are interpolatory and non-rational and that we only use uniform shifts. Our parameterization is simpler than NURBS and thus has several advantages in practical applications such as immediate reconstruction of smooth surfaces by interpolating parameterized point clouds, improved shape modeling by making the process more intuitive, and a simplified formulation of optimization schemes that involve integral- and derivative-dependent quantities. Our framework indicates promising future directions aiming at extending it to a richer family of topologies.

Acknowledgements

This work was funded by the Swiss National Science Foundation under Grant 200020-162343. We are greatful to Zsuzsanna Püspöki for help with the figures and to Irina Radu for the help with the video.

References

AGUET, F., JACOB, M., AND UNSER, M. 2005. Three-dimensional feature detection using optimal steerable filters. In *Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP'05)*, vol. II, 1158–1161.

BOTSCH, M., KOBBELT, L., PAULY, M., ALLIEZ, P., AND LEVY, B. 2010. *Polygon Mesh Processing*. AK Peters.

CATMULL, E., AND CLARK, J. 1998. Seminal graphics. ACM, New York, NY, USA, ch. Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes, 183–188.

DELGADO-GONZALO, R., CHENOUARD, N., AND UNSER, M. 2013. Spline-based deforming ellipsoids for interactive 3d bioimage segmentation. *IEEE Transactions on Image Processing 22*, 10 (October), 3926–3940.

DEROSE, T., KASS, M., AND TRUONG, T. 1998. Subdivision surfaces in character animation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '98, 85–94.

DIERCKX, P. 1984. Algorithms for smoothing data on the sphere with tensor product splines. *Computing 32*, 4 (April), 319–342.

DOO, D., AND SABIN, M. 1978. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design 10*, 6, 356 – 360.

MANNI, C., PELOSI, F., AND SAMPOLI, L. 2011. Generalized B-splines as a tool in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering 200*, 58, 867 – 881.

PIEGL, L., AND TILLER, W. 2010. *The NURBS Book*, second ed. Springer Berlin Heidelberg.

PRAUN, E., AND HOPPE, H. 2003. Spherical parametrization and remeshing. *ACM Trans. Graph. 22*, 3 (July), 340–349.

PRAUTZSCH, H., BOEHM, W., AND PALUSZNY, M. 2002. *Bézier and B-Spline Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

SCHMITTER, D., DELGADO-GONZALO, R., AND UNSER, M. 2015. Trigonometric interpolation kernel to construct deformable shapes for user-interactive applications. *IEEE Signal Processing Letters 22*, 11 (November), 2097–2101.

STAM, J., AND LOOP, C. T. 2003. Quad/triangle subdivision. *Comput. Graph. Forum 22*, 1, 79–86.