Adapted Dynamic Meshes for Deformable Surfaces

Fernando de Goes Felipe P. G. Bergo Alexandre X. Falcão Siome Goldenstein
LIV – IC – UNICAMP
Campinas, SP – Brazil
{fernando.goes,felipe.bergo,afalcao,siome}@ic.unicamp.br

Luiz Velho
VISGRAF Project – IMPA
Rio de Janeiro, RJ - Brazil
lvelho@impa.br

Abstract

Deformable objects play an important role in many applications, such as animation and simulation. Effective computation with deformable surfaces can be achieved through the use of dynamic meshes. In this paper, we introduce a framework for constructing and maintaining a time-varying adapted mesh structure that conforms to the underlying deformable surface. The adaptation function employs error metrics based on stochastic sampling. Our scheme combines normal and tangential geometric correction with refinement and simplification resolution control. Furthermore, it applies to both parametric and implicit surface descriptions. As the result, we obtain a simple and efficient general scheme that can be used for a wide range of computations.

1. Introduction

Three-dimensional models are fundamental in all areas of computer graphics. The traditional way to represent them is through polygonal meshes. Good meshes are needed for applications ranging from visualization, modeling, simulation and animation. As the complexity of the model increases, the mesh becomes more detailed. Large meshes are often redundant and inefficient for computation. The solution for this problem is to use adapted meshes.

Mesh adaptation is an area of intense research because of its importance. However, most of the work in this area has concentrated on adapted meshes for static models. The problem of creating dynamic meshes for deformable objects received less attention up to now. Although part of the results obtained for adaptation of static meshes can be used in the case of dynamic meshes, truly effective computation can only be achieved by exploiting the specific nature of deformable objects.

In this paper, we propose a framework for generating adapted dynamic meshes for deformable surfaces. We construct an initial mesh by refinement of a coarse base mesh. During the simulation, the current mesh is maintained adapted to the underlying deformable surface by local modifications to the mesh geometry and structure. The adaptation is based on the evolution of the surface. Nonetheless, it treats the simulation process as a "black box", since we estimate the dynamics from stochastic samples that are evaluated at each time-step. Furthermore, our scheme is general and can be applied to both parametric and implicit surfaces alike.

The main contributions of our work are as follows:

- **General Framework**: we develop a complete framework for adaptation of dynamic meshes that approximate a time-varying deformable surface. The resulting meshes exhibit very good properties, in terms of mesh quality (i.e., small number of faces, well shaped elements, semi-regular structure, graded variable resolution, bounded connectivity) and temporal coherence (i.e, it evolves in time through geo-morphing).
- **Stochastic Sampling**: we employ stochastic stratified sampling to estimate the error between the mesh approximation and the true surface in a robust and efficient manner. Additionaly, we formulated a flexible error metric that can take into account both geometric and other aspects of the adapted mesh.
- **Resolution Control**: we build on the infra-structure provided by semi-regular 4-8 meshes to create an efficient mechanism for dynamic mesh resolution adaptation using scheduled simplification and refinement stellar operations.

- **Geometric Quality**: we extend the intrinsic Laplacian mesh smoothing to incorporate curvature sensitive behavior as the surface deforms. In this way, we are able to maintain a triangulation that follows surface features with controlled aspect ratio.

The paper is structured as follows: in Section 2 we give an overview of work related to adapted dynamic meshes; in Section 3 we describe the main components of the proposed framework; in Section 4, we show examples of the application of our framework to various deformable surfaces; and finally in Section 5 we conclude with an evaluation of the results and a discussion of future work.

2. Related Work

Related work falls under the categories of topological data structures for mesh representation, frameworks for mesh adaptation and applications using deformable mesh surfaces.

There has been extensive research in multiresolution mesh representations [13]. This type of structure can be constructed from a dense mesh using simplification [14] or from a coarse mesh using refinement [20].

Variable resolution schemes for mesh adaptation can be developed using local simplification and refinement operators [19]. These schemes serve as the foundation of mesh adaptation frameworks, such as progressive meshes [15] and 4-K meshes [23].

Most of the work in mesh adaptation has been applied to static surfaces. Some relevant papers in this category are [26] and [5].

An important related application that exploits adapted meshes is the view-dependent visualization of terrain data [9], [18].

The research on adaptive meshes for deformable surfaces usually is combined with the simulation process. Some examples are the pioneering work of Welch et al. for variational modeling [25], the work of Bowden et al. for medical applications [4], and more recently [8].

Recent work that uses multiresolution meshes for deformable surfaces include [16] and [6].

General mechanisms for dynamic mesh adaptation of deformable surfaces have been proposed by [17] and [21]. These mechanisms employ edge-based topological data structures together with local mesh operators, such as the ones supported by the libraries CGAL [10] and OpenMesh [3].

Here, we use the stellar mesh library [22] as the basis for our dynamic mesh adaptation framework of deformable surfaces. However, we enhance this infra-structure for the case of time-varying repesentations.

3. The Adaptation Framework

There are several criteria to determine the quality of a polygonal representation of a continuous surface. First, the piecewise linear approximation given by the mesh should be within a tolerance (according to some *error metric*). Second, the *mesh size* (i.e., the number of elements) should be small. Third, the *shape* of polygons (i.e., aspect ratio, orientation) should be bounded and adapted to surface features. Fourth, the *structure* of the mesh (i.e., degree of vertices) should be as regular as possible – which in the case of triangle meshes mean valences close to six.

Note that the criteria above are interdependent and may conflict with each other. For example, a large number of elements could be required for accurate approximations. In essence, the construction of a good mesh can be posed as a constrained optimization problem. One strategy to solve such a problem is through a mesh adaptation process that balances the different criteria according to restrictions.

The adaptation process controls mesh geometry and topology to achieve its goals. Geometric control determines the displacement of vertex positions – normal or tangential to the surface. Topology control determines vertex connectivity by structural operations that may also affect the resolution of the mesh, such as vertex insertion and removal.

Our adaptation framework combines structural and geometric operations in order to maintain the quality of a dynamic mesh while the underlying surface is deforming. It keeps the surface approximation under some prescribed tolerance using a small number of elements that are well shaped and aligned to features. Furthermore, the resulting mesh structure has bounded and graded regularity.

An overview of the whole process is as follows: Given a deformable surface $S(t)$ and a base mesh M that has the same topology as S, we assume the ability to sample points on S at each time step t of the simulation, and that S does not change topology. The adaptation is an iterative algorithm: it has an initialization phase and a main loop. In the initialization, the base mesh is adaptively refined to create a mesh M_0 approximating S at $t = 0$. In the loop, for each time step $t = i$ of the simulation, the current mesh M_i is adapted by structural and geometric operations using a sampling of S_i.

As the model deforms, we apply the following two-step procedure:

Structural Operations – Simplification / Refinement:
First the surface is sampled at the current time step. Then, the mesh resolution is modified based on the error criteria. We remove vertices in regions with error below the threshold and subdivide edges in regions with error above it[1].

Geometric Operations – Vertex Position Correction:
First, new vertices are projected onto the surface and exis-

[1]Note that the error metric is usually based on geometric approximation, but can also incorporate other application-specific criteria.

ting vertices updated for the current time step. Then, all vertices are displaced on the tangent plane of S using the local curvature of the surface.

3.1. Stochastic Sampling and Error Metrics

In order to determine the correct level of approximation we need to evaluate an integral of the error over surface regions. Exact computation might be prohibitive, so, an alternative is *stochastic sampling*. Stochastic sampling selects points in the region of interest, and keeps the density of samples per area constant with uniform distribution. The quality of the integral's approximation (the sum of the samples' errors) is related to the sampling density.

This strategy has the advantage of being independent from the surface description. All it requires is the computation of sample points on the surface, which can be done easily in most cases. Stochastic sampling also provides a flexible mechanism to estimate different error metrics and to learn the dynamics of the deformation. In our experiments we have adopted an L^∞ norm.

Sampling is performed in the initialization phase and updated at each time step of the simulation. An stratified pattern (in barycentric coordinates) is kept for this purpose on the triangles of the mesh with a density prescribed by the application. (See Figure 1(b)).

3.2. Structural Operations

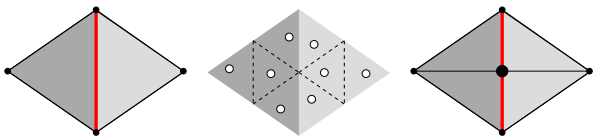

Figure 1. split $(a) \rightarrow (c)$ and weld $(c) \rightarrow (a)$ of a mesh region are based on error computed using a stratified sampling pattern (b).

We employ the stellar operations edge *split* and *weld* to change the structure of the mesh. Their action is respectively to refine / coarsen a basic region of the mesh by subdividing two triangles that share an edge / simplifying four triangles incident to a vertex (See Figure 1).

The mesh representation has an underlying semi-regular 4-8 structure [22]. As a consequence, the mesh structure adaptation guarantees a bounded connectivity of the mesh. That is, if the degree of vertices in the base mesh is limited in the interval $[d_{\min}, d_{\max}]$, the structural operations change the mesh resolution by creating new vertices with valences inside this interval. In the case of regular 4-8 meshes, the vertex valence stays between 4 and 8 – the average is equal to 6, the optimal degree for regular meshes.

We associate the error over basic mesh regions with split edges and weld vertices. The set of these regions covers the entire mesh (See [22]). Also, they play a symmetric role for mesh adaptation – one is used for refinement and the other for simplification. To make the adaptation criteria consistent, we compute the error e from a stochastic sampling over the region, using e with a threshold T for refinement and simplification.

Split edges and weld vertices are kept on two priority queues ordered by the error. At each time step (t_i) of the simulation, we first update the error from the current position of sample points. Then, we simplify the mesh by removing all vertices with an associated error below $T - \varepsilon$. After that, we refine the mesh by subdividing all edges with associated error above $T + \varepsilon$. In this way, the mesh approximation stays inside a tubular region around the true surface. Note that we include a tolerance interval of 2ε around the threshold T in order to deal with hysteresis and avoid undesired instability in the resolution adaptation.

3.3. Geometric Operations

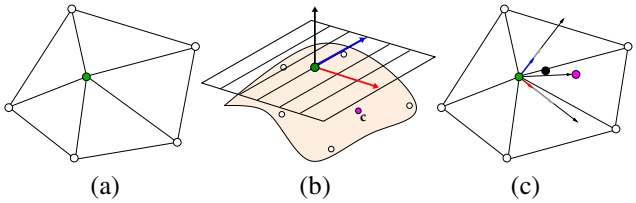

Figure 2. (a) Polygon mesh. (b) Tangent plane and principal curvatures. (c) Curvature-based vertex correction.

During the structural adaptation, the vertex positions are set to reflect the current state of the deformable surface. However, after the geometry update there is no guarantee that all the triangles of the mesh will remain well shaped.

We want a mesh with *good parametrization*. This happens when neighbor triangles have edges with similar length. We activate this by performing an intrinsic Laplacian mesh smoothing that is curvature and feature sensitive.

Let's call p a vertex of the mesh, c the centroid of p one-ring neighbor's, $\vec{v} = p - c$ the displacement vector, and $\vec{n}$ the normal vector at p. Intrinsic Laplacian smoothing displace p along the projection of v on the tangent plane at p (along $\vec{d} = \vec{v} - \langle \vec{v}, \vec{n} \rangle \vec{n}$).

We need to keep vertices on the surface features and regions with high curvature. The Laplacian adjustment has to be curvature sensitive, and behave differently along each *principal curvature direction* $\vec{u}_i$, depending on the *principal*

curvature value λ_i. Larger displacement for lower curvatures.

The complete geometry adaptation process displaces each mesh vertex p according to Equation 1, and then reprojects it onto surface (See Figure 2).

$$p = p + \langle \vec{u}_1, \vec{d} \rangle \exp(-k\lambda_1)\vec{u}_1 + \langle \vec{u}_2, \vec{d} \rangle \exp(-k\lambda_2)\vec{u}_2, \quad (1)$$

where u_i are the curvature directions, λ_i are the curvature values, and k is a constant for the curvature sensitive.

The reprojection step depends on the surface description. In case of parametric surfaces the projection is trivial, since we can evaluate the parametrization over the mesh. In the case of implicit surfaces we use the gradient field of the implicit function for the projection.

4. Results and Applications

Our method works with implicit and parametric models. Here, we show examples for both representations with distinct adaptation criteria. First we demonstrate basic adaptation with a deformable implicit cylinder and dynamic parametric height surface. Second, we show an implicit sphere with variable radius that uses multiple adaptation criteria. Finally, we present a real medical application that is currently an aiding tool for diagnosis of *dysplastic lesions* in the brain.

4.1. Implicit Cylinder

An implicit cylinder has an overall uniform curvature, so, a regular 4-8 mesh is a good surface approximation (left of Figure 3). The shades of green represent the mesh approximation error, yellow means no error, and red means an error beyond the threshold. As we bend the cylinder, there are areas along deformations that have more curvature and detail. We use the geometrical error to decide when and where to refine, creating new vertices, edges, and faces. This result is also on the right of Figure 3. Note the overall constant approximation error.

4.2. Parametric Height Functions

Height functions are a family of parametric surfaces. The x,z coordinates act as the parameters to the height function $y = f(x,z)$.

In Figure 4, we show how the mesh adapts appropriately to deforming height function using the geometric criteria. In Figure 5, we animate the deformable *sinc* height function

$$y = \alpha \ \mathrm{sinc}(\beta r) \ \exp(\kappa r), \quad (2)$$

where $r = \sqrt{(x^2 + z^2)}$ is the distance to the origin, β and κ are shape parameters, and α is the deformation parameter of the surface. The upper row of Figure 5 shows snapshots of the surface's deformation rendered with Gouraud shading. The lower row shows how the mesh is suited to the surface's curvature, viewed from the top and rendered using pseudocolors indicating the value of the highest curvature. The curvature-sensitive adjustment plays a crucial role to keep the ridges of the *sinc* function sharp and well defined, while the error criteria controls the adaptation everywhere else.

4.3. Multi-criteria Sphere

Here, isosurfaces of an implicit sphere demonstrate how our method copes with different types of adaptation simultaneously. The left hemisphere uses the geometry error, the right uses a fixed edge size. Figure 6 shows the behavior of the mesh for radii of 0.6, 1.0, 1.6, and 9.0. Dark gray means higher curvature. The lower row has the isosurfaces scaled to a common size. Small radii isosurfaces have high curvature, so their edge size criteria generates coarser meshes than the geometric error. Larger radii have smaller curvatures, and the geometric error criteria leads to coarser meshes.

4.4. Twisted Surface

Now, we present an implicit surface modified by a twist deformation. The surface is the extrusion of a L^p curve, defined implicitly as $y^p + z^p = r^p$, where p defines the metric used on a circular domain, and r is the radius of this circunference. The twist deformation rotates each transversal section of the object by an increasing angle according to its x coordinate. At each time step, the rate of rotation is increased, and the surface becomes closer to a screw (See Figure 8).

4.5. MRI: Discrete Volumetric Data

Finally, we show a real medical application of our approach. Based on discrete volumetric brain MR data, we perform volume segmentation and calculate an implicit representation of isolayers of the brain based on a distance transform. By varying dynamically the isovalue we obtain a family of isosurfaces that can be used to analyze the MR data. This tool allows medical professionals to visualize brain MR intensities as concentric well-adapted surfaces. It is easier to detect dysplastic lesions through inspection of these isosurfaces, instead of the orthogonal cuts available in conventional exams [1]. Work is under way for automatic detection of lesioned areas. In Figure 7, we show the mesh/error and fully rendered version of two different isosurfaces.

5. Conclusions and Ongoing Work

We introduced a complete framework to dynamically adapt meshes that approximate deformable surfaces with good properties. Our method is based on a two step adaptation scheme defined by structural and geometric operations.

We also developed a flexible adaptation scheme that supports a combination of different criteria. The geometric approximation is estimated through stochastic sampling of the deformable surface, which makes possible to employ various error metrics.

Stochastic sampling is a powerful technique. We are currently working on its use for dynamic learning and adaptation of mesh deformations. We are also investigating new operators that will allow the mesh to dynamically change topology as it deforms.

6. Acknowledgments

This research has been sponsored by FAPESP (Proc. 06/50980-8 and 03/13424-1), CNPq (Proc. 302427/04-0), FAPERJ, FINEP, CAPES, and IBM Brazil.

References

[1] F. P. G. Bergo and A. X. Falcão. Fast and automatic curvilinear reformatting of MR images of the brain for diagnosis of dysplastic lesions. In *IEEE International Symposium on Biomedical Imaging*, 2006.

[2] G. Berti. *Generic software components for Scientific Computing*. PhD thesis, BTU Cottbus, 2000.

[3] M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt. Openmesh - a generic and efficient polygon mesh data structure. In *OpenSG PLUS Symposium*, 2002.

[4] R. Bowden, T. Mitchell, and M. Sahardi. Real-time dynamic deformable meshes for volumetric segmentation and visualization. *Proc. BMVC*, 11:310–1=319, 1997.

[5] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, Aug. 2004.

[6] C. DeCoro and S. Rusinkiewicz. Pose-independent simplification of articulated meshes. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 17–24, New York, NY, USA, 2005. ACM Press.

[7] T. Dey, H. Edelsbrunner, and S. Guha. Computational topology. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry (Contemporary mathematics 223)*, pages 109–143. American Mathematical Society, 1999.

[8] Y. Duan, J. Hua, and H. Qin. Interactive shape modeling using lagrangian surface flow. *Visual Comp.*, 21(5):279–288, 2005.

[9] M. A. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein. Roaming terrain: Real-time optimally adapting meshes. In *IEEE Visualization '97*, pages 81–88, Nov. 1997.

[10] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schonherr. On the design of CGAL a computational geometry algorithms library. *SP&E*, 30(11):1167–1202, 2000.

[11] L. D. Floriani, P. Magillo, and E. Puppo. Efficient implementation of multi-triangulations. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *IEEE Visualization '98*, pages 43–50, 1998.

[12] L. D. Floriani, E. Puppo, and P. Magillo. A formal approach to multiresolution modeling. In W. S. er, R. Klein, and R. Rau, editors, *Theory and Practice of Geometric Modeling*. SpringerVerlag, 1996.

[13] M. Garland. Multiresolution modeling: Survey & future opportunities. In *Eurographics, State of the Art Report*, 1999.

[14] M. Garland and Y. Zhou. Quadric-based simplification in any dimension. *ACM Trans. on Graphics*, 24(2):209–239, 2005.

[15] H. Hoppe. Efficient implementation of progressive meshes. *Computers & Graphics*, 22(1):27–36, February 1998.

[16] S. Kircher and M. Garland. Progressive multiresolution meshes for deforming surfaces. In *2005 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 191–200, July 2005.

[17] L. P. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. *Computer Graphics Forum*, 19(3), 2000.

[18] P. Lindstrom and V. Pascucci. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):239–254, July - September 2002.

[19] E. Puppo. Variable resolution triangulations. *Computational Geometry Theory and Applications*, 11(34):219–238, 1998.

[20] P. Schroeder. Subdivision for modeling and animation, 1998. Course notes of Siggraph 98. ACM SIGGRAPH.

[21] A. Shamir, C. L. Bajaj, and V. Pascucci. Multi-resolution dynamic meshes with arbitrary deformations. In *IEEE Visualization*, pages 423–430, 2000.

[22] L. Velho. A dynamic adaptive mesh library based on stellar operators. *Journal of Graphics Tools*, 9(2):1–29, 2004.

[23] L. Velho and J. Gomes. Variable resolution 4-k meshes: Concepts and applications. *Comp. Graphics Forum*, 19:195–212, 2000.

[24] L. Velho and D. Zorin. 4-8 subdivision. *Computer-Aided Geometric Design*, 18(5):397–427, 2001. Special Issue on Subdivision Techniques.

[25] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In *SIGGRAPH*, pages 247–256, 1994.

[26] Z. J. Wood, M. Desbrun, P. Schröder, and D. Breen. Semi-regular mesh extraction from volumes. In *IEEE Visualization 2000*, pages 275–282, Oct. 2000.

Figure 3. Geometric error as refinement criteria: bending of implicit cylinder.

Figure 4. Dynamic adaptation of parametric surface.

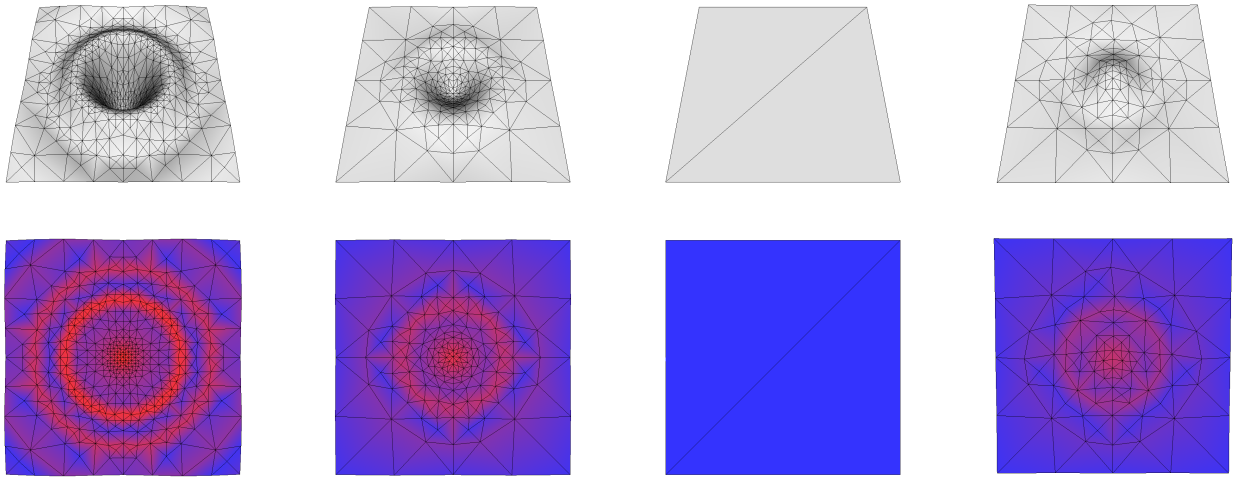

Figure 5. Deformations of a height function. Upper row: perspective view. Lower row: adaptation over the curvature.

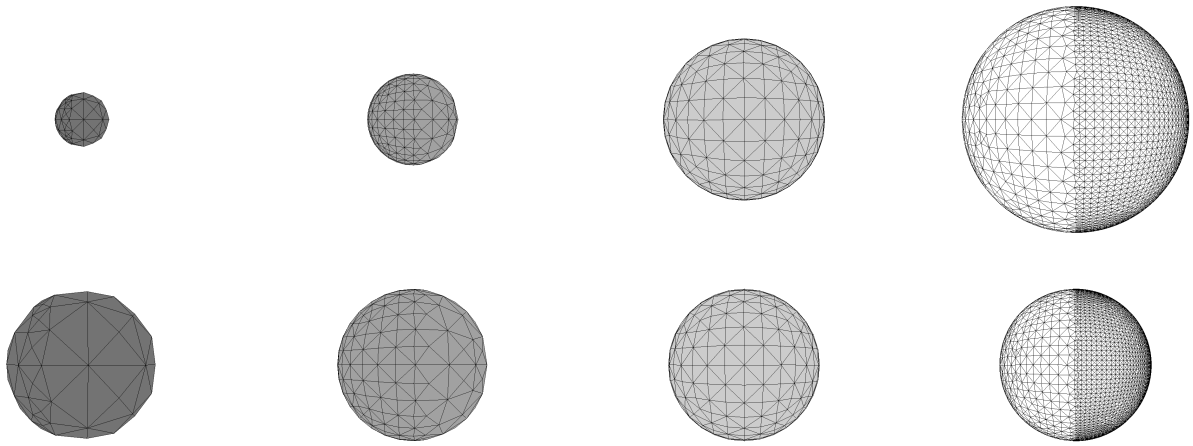

Figure 6. Multi-criteria for adaptation of several isosurfaces of an implicit sphere.

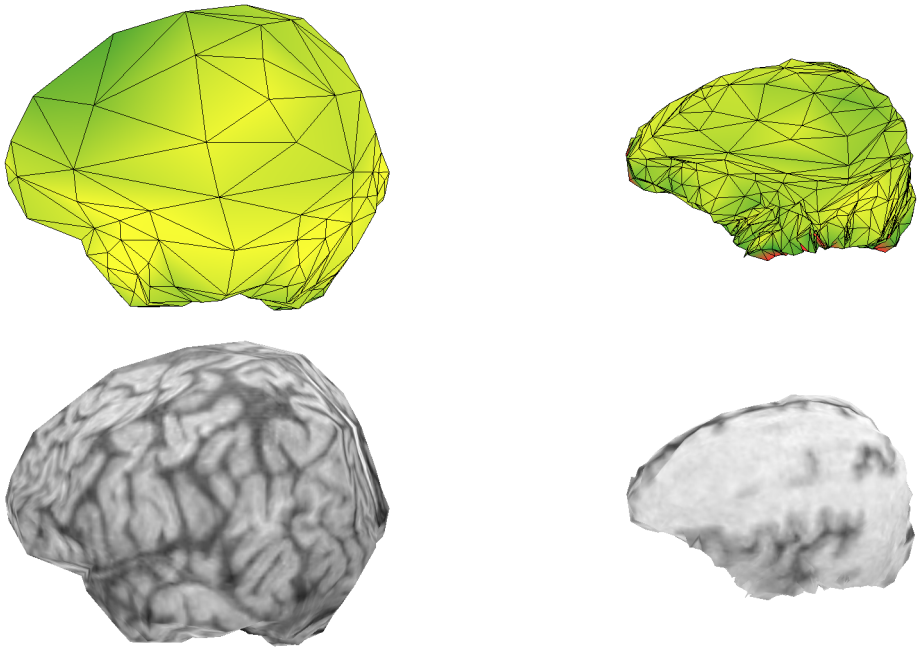

Figure 7. Example of a dysplasia diagnostic tool. MRI provides discrete volumetric data.

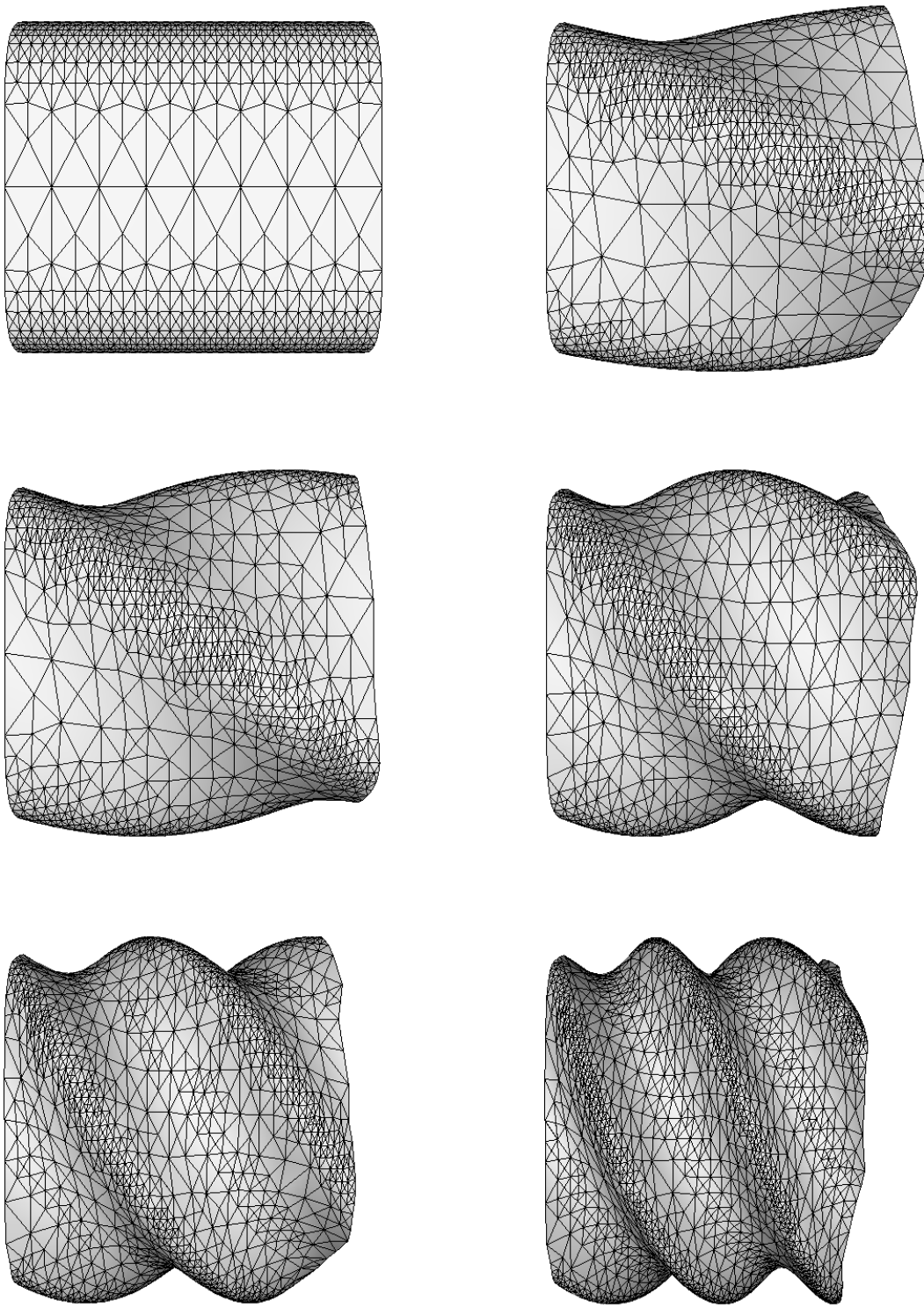

Figure 8. Implicit twisted surface.