# A Simple and Flexible Framework to Adapt Dynamic Meshes

Fernando de Goes [a,*] Siome Goldenstein [a] Luiz Velho [b]

[a] *Instituto de Computação - Universidade Estadual de Campinas - Caixa Postal 6176, 13083-971, Campinas, SP, Brazil*
[b] *IMPA - Instituto de Matemática Pura e Aplicada - Estrada Dona Castorina, 110, 22460, Rio de Janeiro, RJ, Brazil*

**Abstract**

Many graphics applications represent deformable surfaces through dynamic meshes. To be consistent during deformations, the dynamic meshes require an adaptation process. In this paper we present a simple and flexible framework to adapt dynamic meshes following deformable surfaces. Our scheme combines normal and tangential geometric corrections with refinement and simplification resolution control. It works with different surface descriptions, and supports application-specific criteria. We also introduce a stochastic sampling approach to measure the geometric error approximation. As an example, we couple our framework with numeric simulations, such as a particle level-set method.

*Key words:* mesh adaptation, dynamic meshes, deformable surfaces

## 1. Introduction

Triangulated meshes are a simple and fundamental tool to represent complex geometries and 3D objects for graphics applications ranging from animation, visualization, and modeling. The simplicity of a mesh consists on a pair of structural and geometric data. To achieve desired results, a mesh must present good properties, such as a small number of faces, well shaped elements, and a semi-regular structure.

As the complexity of the surfaces increases, guaranteed mesh quality becomes a challenge task. Mesh adaptation is the procedure which intends to improve mesh quality applying structural and geometric corrections. For static models, there is an extensive set of adaptation techniques, including multiresolution and remeshing approaches. However, the problem of adapting dynamic meshes received less attention up to now. Although part of the results obtained for static meshes can be used for dynamic meshes, truly effective computation can only be achieved by exploiting the specific nature of deformable objects.

In this paper, we propose a simple and efficient framework to generate adapted dynamic meshes for deformable surfaces. Our method is *flexible*: coupling application-specific criteria into the adaptation process, and treating parametric or implicit, continuous or discrete objects. During the simulation, we maintain the mesh adapted by structural operations based on stellar theory, and adjusting vertices positions to preserve surface features. We also present a stochastic sampling as an accurate and easy to control geometric error metric. We enhance our framework and error metric to track explicit boundaries evolved by numeric simulations. These flows require an intense mesh adaptation, expanding and contracting the surface in different regions. For simplicity, we assume that the surface topology does not change.

The main contributions of our work are as follows:

⋆ Expanded version of the article *Adapted Dynamic Meshes for Deformable Surfaces* presented at the XIX Brazilian Symposium of Computer Graphics and Image Processing – SIBGRAPI (Brazil, October 2006).

* Corresponding author.
*Email addresses:* fdegoes@liv.ic.unicamp.br (Fernando de Goes), siome@ic.unicamp.br (Siome Goldenstein), lvelho@impa.br (Luiz Velho).

- **Simplicity**: we develop a simple 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 and temporal coherence.
- **Flexibility**: our framework supports different types of surface descriptions and deformation simulations. It also supports application-specific criteria in the adaptation process.
- **Resolution Control**: we build on the infrastructure 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.
- **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.

The remainder of the paper is organized as follows. Section 2 gives an overview of related work on adapted dynamic meshes. Section 3 describes the details of our framework. Section 4 explains the stochastic sampling approach to estimate geometric approximation error. Section 5 presents examples of our method into various deformable surfaces, including fluid simulations. Finally, Section 6 concludes with an evaluation of the results and discussion of future work.

## 2. Related Work

A substantial amount of work has addressed mesh adaptation using multiresolution representation [10]. This type of structure can be built from a dense mesh using simplification [11] or from a coarse mesh using refinement [26]. Variable resolution schemes [25] present a hybrid method to adapt meshes using local operations for simplification and refinement simultaneously, and serve as the foundation of adaptation frameworks, such as progressive meshes [14] and 4-K meshes [30]. Here, we use a semi-regular 4-8 mesh structure based on stellar operations [29].

Most of the applications of multiresolution schemes are related with static meshes. Examples are view-dependent visualization of terrain data [8, 20] and mesh parameterization [19]. Nonetheless, some works extend these schemes to treat dynamic meshes. Shamir et al. [27] presented an adaptive multiresolution representation called T-DAG, which involves different poses of a model in the same structure and treats arbitrary connectivity and topology changes. A progressive representation for dynamic meshes is given in [17], creating a time-varying multiresolution hierarchy from a initial mesh and a sequence of update operations. In [6], a multiresolution structure is built from a dense and skeletally articulated mesh and a probability function of potential poses. Goldenstein et al. [12] pre-processed dynamic meshes with a semi-regular 4-8 mesh structure to compute a final mesh with low geometric error over a set of possible deformations. This mesh was used to track face dynamics in video sequences. Note that all these approaches require a priori knowledge either of the possible deformations or of the resulted deformed meshes.

Remeshing techniques are another way to adapt meshes constructing brand new meshes from input ones. These methods provide a good user-control over the quality of the final mesh, varying the density of points and the alignment to surface features. Some relevant papers in this category are [1, 2, 5, 28]. Despite of excellent adaptation of static models, remeshing can become very expensive if applied at each deformation of a dynamic mesh.

Mesh editing has also presented adaptation techniques to treat user interactions. In [18], a detail function is derived from multiresolution representations to adapt mesh connectivity during its modification. In [7], meshes are adapted based on Lagrangian surface flows from distance fields defined by the user.

In computational fluid dynamics, the explicit representation of fluid boundaries is a hard problem, and provides many benefits during simulations as noted in [3, 15]. Similar to particle-based Lagrangian methods, evolving meshes confronts difficulties to ensure good sampling of the surface, no self-intersections, and to handle degenerate cases. Jiao et al. [16] presented an anisotropic adaptation optimization to control evolving meshes. The method combines edge operations and an extension of the quadric-based surface analysis [13] to redistribute vertices and guarantee geometric accuracy. In [4], a complete framework is proposed to track explicit surfaces, treating topology changes and adapting triangulations through vertex insertion and removal.

## 3. The Adaptation Framework

In this section, we explain each step of our framework. First, we give an overview of the whole process and show its flexibility. After, we present details of the structural and geometric operations.

### 3.1. *Overview*

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., 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.

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.

However, when the underlying surface is deforming, the adaptation process becomes even more complicated. The deformation dynamics adds new constraints to the adaptation process. During the deformation, the dynamic mesh should present temporal coherence, while maintains the mesh quality.

Our adaptation framework works on dynamic meshes through the combination of structural and geometric operations at each time step of the deformation. 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.

To adapt dynamic meshes, we assume the definition of three "black boxes" given by the application. First of all, the application must define a surface description, which supports the sampling of points on the surface. The application must also describe the deformation process, computing displacement vectors for each point on the surface. Note that these displacement vectors represent a first order knowledge about the dynamics process. At last, the application must specify an error metric to guide the adaptation process.

Therefore, given a deformable surface $S(t)$ which does not change topology, and an initial mesh $M$ that has the same topology as $S$, our adaptation process works as an iterative algorithm, i.e., it has an initialization phase and a main loop. In the initialization, the initial mesh $M$ is converted to a semi-regular 4-8 mesh $\tilde{M}$, and then refined to create a mesh $M_0$ approximating $S$ at $t = 0$. In the loop, at each time step $t = i$ of the simulation, the current mesh $M_{i-1}$ is first deformed by the displacement vectors into the mesh $M_i$. After that, the mesh $M_i$ is adapted by structural and geometric operations using a sampling of $S(i)$.

It is worth to note that the initial mesh $M$ may not be a semi-regular 4-8 mesh. We convert it into a semi-regular 4-8 mesh $\tilde{M}$ using the algorithm proposed in [29]. This technique creates $\tilde{M}$ from any triangulated mesh $M$ with any connectivity. It also guarantees a bounded structure for $\tilde{M}$, that is, if the degree of the vertices of $M$ is limited by the interval $[d_{min}, d_{max}]$, then the converted mesh $\tilde{M}$ also has vertices which degrees are inside this interval.

### 3.2. *Structural Operations*

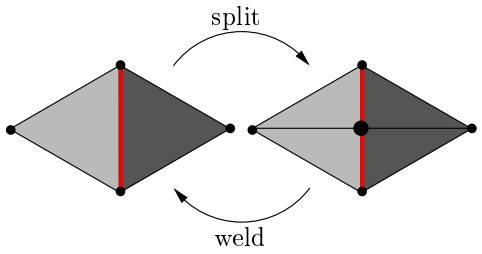


Fig. 1. Stellar operations on a mesh region.

We employ the stellar operations *edge split* and *vertex 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 (Figure 1).

We have already pointed out that the mesh representation has an underlying semi-regular 4-8 structure. As a consequence, the mesh structure adaptation guarantees the same bounded connectivity of

the initial mesh $M$ for all meshes $M_i$ created during the simulation. Thus, the structure of the initial mesh $M$ is preserved along the deformation. Furthermore, we do not simplify vertices of the base 4-8 mesh $\tilde{M}$, so the mesh $\tilde{M}$ indicates the coarsen resolution of the adaptation process.

We associate the adaptation error over basic mesh regions with split edges and weld vertices. The set of these regions covers the entire mesh (see [29]). 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 the application metric over each mesh region, and use $e$ 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 the mesh. 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 $2\varepsilon$ around the threshold $T$ in order to deal with hysteresis in the resolution adaptation, such as pumping artifacts. In all experiments, we set $\varepsilon$ to $10^{-6}$.

### 3.3. *Geometric Operations*

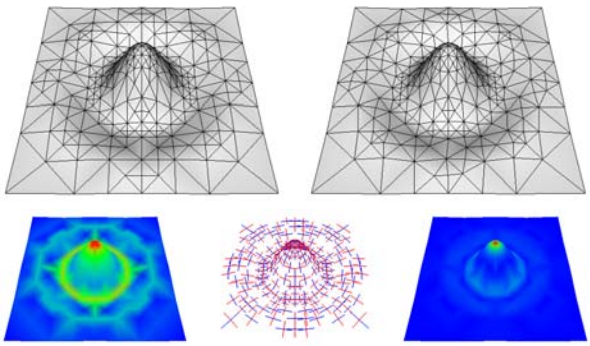

Fig. 2. Geometric operation on a parametric height function. First row: surface before and after geometric operation. Second row: absolute maxima curvatures, principal directions, and absolute minima curvatures. Colors indicate curvature values arising from blue to red.

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

We want a mesh with a *good parameterization*. This happens when neighbor triangles have similar-length edges. We ensure 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 neighbors, $\vec{v} = p - c$ the offset vector, and $\vec{n}$ the normal vector at $p$. Intrinsic Laplacian smoothing displaces $p$ along the projection of $\vec{v}$ on the tangent plane at $p$ (i.e., 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 its *principal curvature value* $\lambda_i$. For this purpose, we weight the displacement on each direction $\vec{u}_i$ according to the absolute value of $\lambda_i$. Larger weights for lower curvatures.

To identify features and regions of high curvature, we assume that the curvature is well defined and continuous on the surface, and then we normalize its absolute values using its minimum $\lambda^-$ and the maximum $\lambda^+$ absolute values.

We also define a parameter $k$ to provide application control of the sensitivity to features on the adaptation process. High values of $k$ indicate less sensitivity to features and an isotropic adaptation. Low values of $k$ indicate high sensitivity to features and an anisotropic adaptation.

Therefore, the geometry operation displaces each mesh vertex $p$ according to

$$p = p + w(|\lambda_1|)\langle \vec{u}_1, \vec{d} \rangle \vec{u}_1 + w(|\lambda_2|)\langle \vec{u}_2, \vec{d} \rangle \vec{u}_2$$

$$w(x) = \begin{cases} \exp\left(-k\left(\dfrac{x - \lambda^-}{\lambda^+ - \lambda^-}\right)\right) & , \lambda^- \neq \lambda^+ \\ 1 & , \lambda^- = \lambda^+ \end{cases} \quad (1)$$

After displace each vertex on its tangent plane, we reproject it onto the surface. This reprojection step depends on the surface description. In case of parametric surfaces, the projection is trivial, since we can evaluate the parameterization over the mesh. In the case of implicit surfaces, we use the gradient field of the implicit function for the projection.

The surface description must also define the computation of its principal curvature values and directions. So the current mesh is not used to compute curvatures. In a special case, if the surface is complex and the computation of its curvatures is diffi-

cult, the application can estimate the curvature values and directions from the mesh [21].

The geometric operation must be applied until the vertices' positions converge or until the aspect ratio of the mesh's triangles achieves an application-specific value. In our implementation, the application can set the desired number of iterations of the geometric operation. More iterations result in a mesh with a better alignment to features.

Figure 2 shows a mesh approximating a parametric height function before and after geometric operation. The curvature-sensitive adjustment plays a crucial role to keep the feature tips and to align mesh triangles into the concentric circumferences of the surface, while the error criteria controls the adaptation everywhere else.

## 4. Stochastic Sampling and Error Metrics

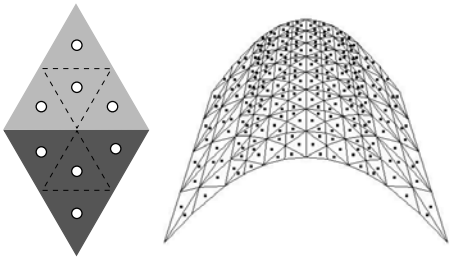

Fig. 3. Stratified sampling pattern on mesh regions.

Despite of different application-specific criteria, the commom metric required for mesh adaptation is the geometric approximation error between the mesh and the underlying surface. In order to evaluate the correct level of approximation, we need to integrate the error over surface regions. The computation of this integral depends on the surface complexity. An alternative and efficient method to estimate the geometric error independently of the surface complexity is the *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. All that this strategy requires is the sampling of points on the surface, which is already provided by our surface description.

In our experiments, we sample mesh regions in the initialization phase and updated them at each time step of the simulation. For this purpose, we keep a stratified pattern (in barycentric coordinates) on mesh triangles with an application-prescribed density (Figure 3).

Stochastic sampling also provides a flexible mechanism to learn the dynamics of the deformation. For instance, we can estimate the deformation velocity and accelaration on a surface region through the analysis of the samples displacements.

Nevertheless, our framework also supports the combination of different error metrics following application-specific critera. In Figure 4, we apply two different criteria to adapt a deformable surface defined by the implicit function $|x|^p+|y|^p+|z|^p = 1$, $p \geq 2$. In the beginning, the surface is a sphere ($p = 2$). We deform it increasing $p$ value. At the left hemisphere, we adapt the mesh using the geometric error estimated by stochastic sampling. At the right hemisphere, we fix edges size. With small values of $p$, the surface is close to a sphere, and has high curvature values. Then, the edge size criteria generates coarser meshes than the geometric error. Increasing $p$, the surface area grows and the high curvatures concentrate on surfaces creases. This leads geometric error criteria to coarser meshes, while the right hemisphere is more refined to preserve the fixed edge size.

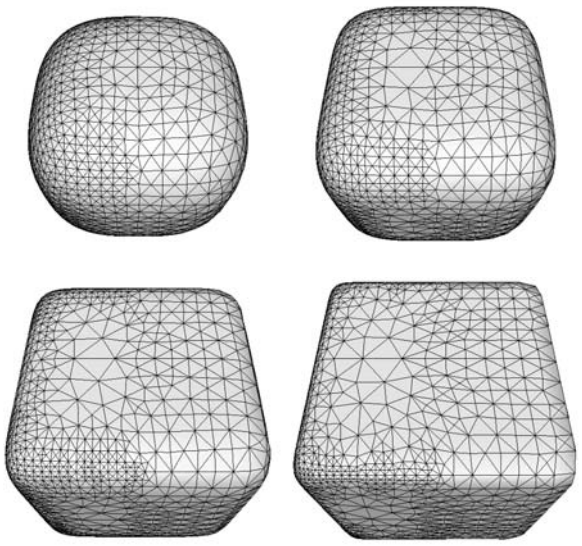

Fig. 4. Multi-criteria adaptation of an implicit surface. The left hemisphere uses the geometric error metric with stochastic sampling, and the right hemisphere fixes edges size.

## 5. Results

We present some preliminary results of our adaptation framework for deformable surfaces. All experiments were carried on a 3.2 GHz Pentium 4 machine with 2 GB RAM. Table 1 reports time statistics. Note that our method works in order of miliseconds, being excellent for real time applications. In all examples, we use only the geometric error criteria estimated with our stochastic sampling approach.

Table 1
Time statistics in miliseconds of our framework. For each example, we measure the initialization time, an average time for deforming and adapting the current mesh, and the total average time. In level-set examples (marked with *), the deforming time also includes the time required by the particle level-set method [9].

| **ms** | *sinc* | *cube* | *twist* | *vortex** | *spiraling** | *2spheres** | *liquid-air* |
|---|---|---|---|---|---|---|---|
| *Init* | 1 | 95 | 200 | 264 | 475 | 566 | 183 |
| *Deform* | 3 | 31 | 97 | 3211 | 3208 | 6543 | 105 |
| *Adapt* | 28.7 | 317 | 673.5 | 290 | 562 | 470 | 265 |
| *Total* | 31.7 | 348 | 770.5 | 3501 | 3770 | 7013 | 370 |

In Figure 5, we present an implicit surface modified by a twist deformation. The surface is the extrusion of a $L^5$ curve, defined implicitly as $|y|^5 + |z|^5 = 1$. 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 with high curvature creases.

We combine our framework with numeric simulations, such as the level-set method. Level-set method [22, 23] is a powerful Eulerian scheme to track and evolve surfaces. The great advantages of this method are the transparency to define surfaces propagations, and the implicit detection and solution of collisions and topology changes. In special, we use the particle level-set method [9] to also minimize volume dissipation, preserving sharp features.

The implict function maintained by the level-set methods describes the surface as the zero-set of a signed distance function. So the level-set methods do not maintain an explicit surface representation, which is desirable in many applications [3, 15]. In another hand, our method provides an explicit surface representation for the evolving front, using the level-set method as a "black box" that describes the surface deformation. As reported in the level-set examples (marked with *) in Table 1, the average time spent to adapt the dynamic meshes is insignificant in comparision with the average time spent to propagate the level-set fronts.

Note that the dynamic mesh maintained by our framework does not participate of the level-set method. Consequently, it does not change the accurarcy of the level-set method.

The next three examples illustrate the coupling of our adaptation process with the particle level-set method [9], using a $100^3$ grid.

First, we use level-set method to deform a sphere of radius 0.15 centered at $(0.5, 0.75, 0.5)$ with a static single vortex velocity field given by:

$$\begin{cases} u(x,y,z) &= \sin^2(\pi x)(\sin(2\pi z) - \sin(2\pi y)), \\ v(x,y,z) &= \sin^2(\pi y)(\sin(2\pi x) - \sin(2\pi z)), \\ w(x,y,z) &= \sin^2(\pi z)(\sin(2\pi y) - \sin(2\pi x)). \end{cases} \quad (2)$$

Figure 6 shows some instants of the adapted surface, using five smoothing iterations at each time step.

We also use the level-set method with the spiraling analytical field from [9], which advects a sphere of radius 0.15 centered at $(0.35, 0.35, 0.35)$ with velocity:

$$\begin{cases} u(x,y,z) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z), \\ v(x,y,z) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z), \\ w(x,y,z) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z). \end{cases} \quad (3)$$

Figure 7 shows results applying 10 smoothing iterations at each time step. In both cases, we modulate the time by the function $cos(\pi t/1.5)$ (see complete animations at supplemental material).

In Figure 8, we use the level-set method and our framework to apply a negative normal flow into the union of two spheres of same radius. Note how the adaptation process controls the resolution structure at the transition between spheres.

Finally, we adapt a liquid-air interface propagated by a particle-based non-Newtonian fluid simulation [24]. The interface and the velocity fields were extrapolated into grids of resolution $120 \times 75 \times 25$, computed separately from our framework. Figure 9 shows various steps of the animation, including the original particles, the adapted mesh, a Gouraud shading, and a realistic rendering.

## 6. Conclusion

In this paper, we present a simple and flexible framework to dynamically adapt meshes that approximate deformable surfaces with good properties. Our method is based on a two step adaptation

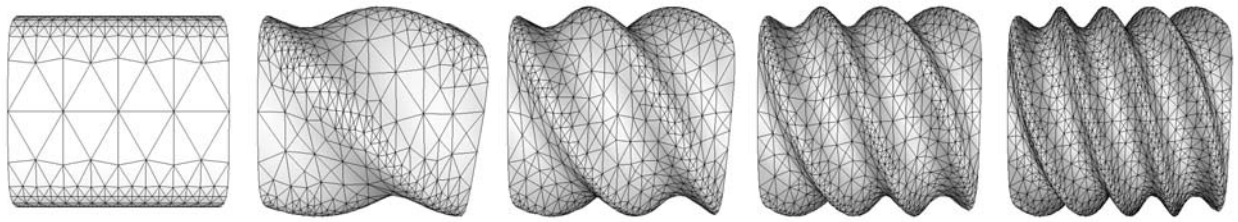

Fig. 5. Implicit twisted surface.

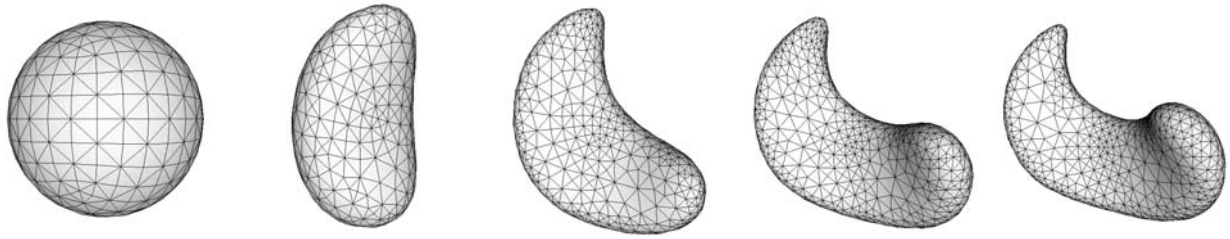

Fig. 6. Adapted meshes of a sphere evolved with a single vortex deformation at iterations 0, 25, 50, 70, and 94, using 0.008 as time step.

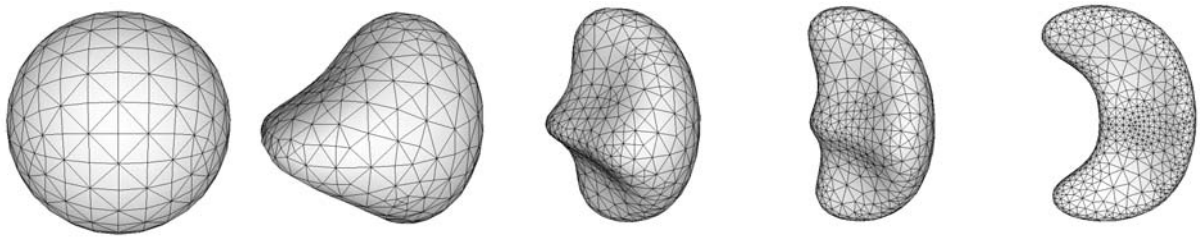

Fig. 7. Adapted meshes generated with the spiraling field [9] at iterations 0, 20, 30, 35, and 63, using 0.012 as time step.

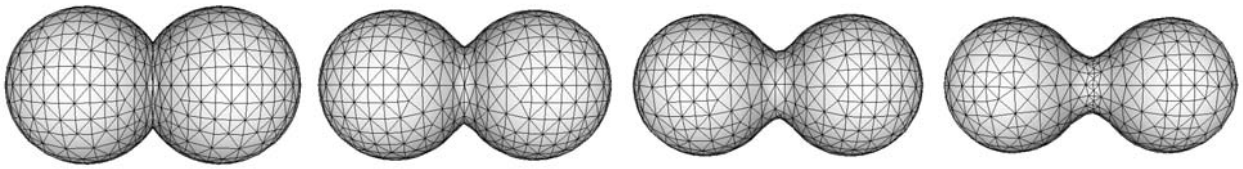

Fig. 8. Negative normal flow on the union of two spheres.

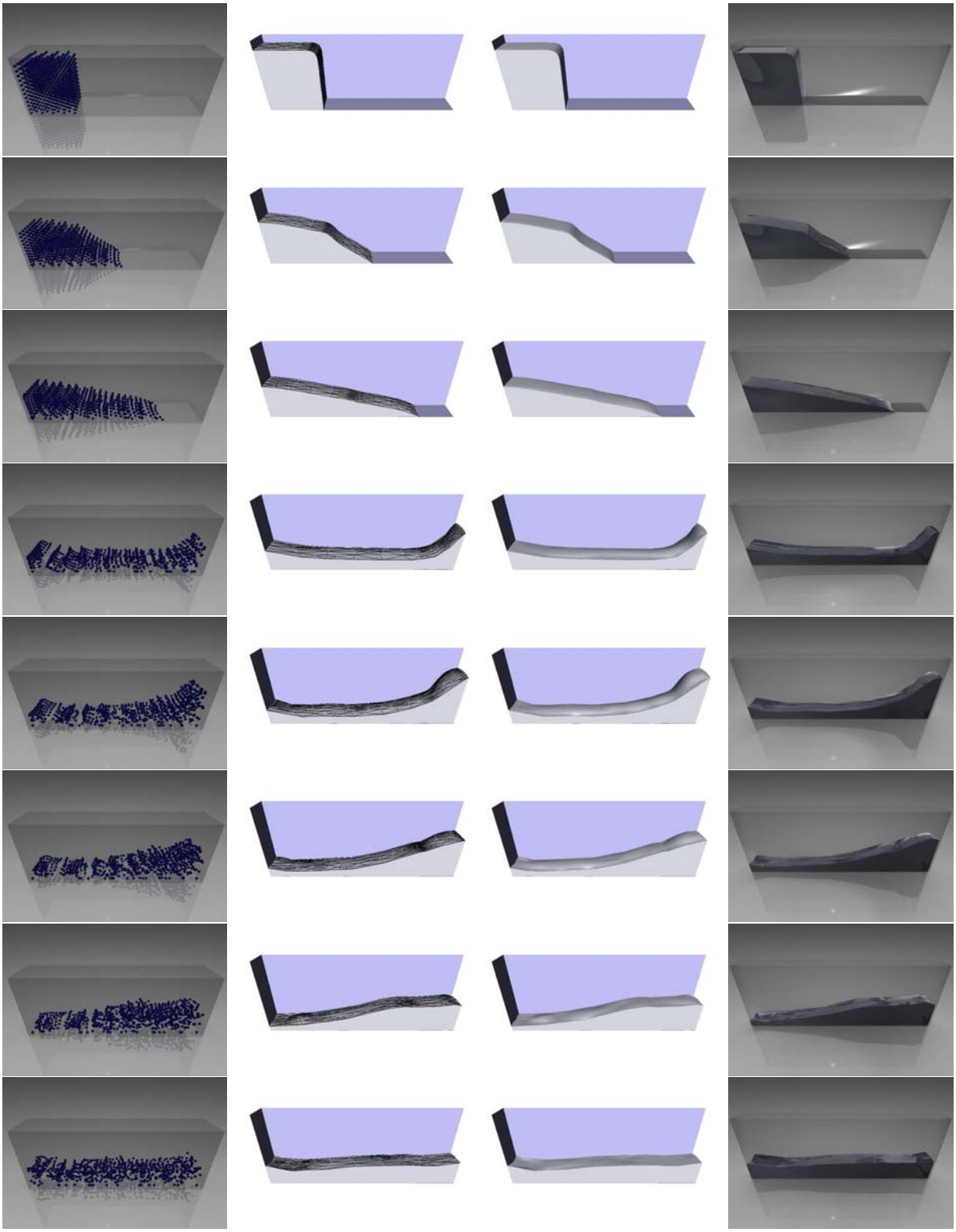

Fig. 9. Adapted meshes for the liquid-air interface propagated by [24]. Each row shows a different step of the animation. Each column shows the original particles, the adapted mesh, a Gouraud shading, and a realistic rendering.

scheme defined by structural and geometric operations. We develop support to different application-specific criteria, and introduce a stochastic sampling approach to estimate geometric error approximation.

We apply our framework to various types of surfaces and deformations, including parametric and implicit examples. In particular, we combine it with dynamic surfaces arising from numeric simulations, such as the particle level-set method [9] and the particle-based non-Newtonian solver [24].

A number of issues can be explored with our framework. First of all, we intend to incorporate new mesh operators to treat topology changes. Future work also involves the reuse of our framework to adapt mesh during user editing, and to increase level-set method accuracy from the adapted meshes.

## Acknowledgments

This work was supported by FAPESP (Proc. 06/58007-7), CNPq, FAPERJ, FINEP, and IBM Brasil. The authors would like to thank Afonso Paiva for the liquid-air example data.

## References

[1] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, M. Desbrun, Anisotropic polygonal remeshing, ACM Trans. Graph. 22 (3) (2003) 485–493.

[2] P. Alliez, E. C. de Verdiere, O. Devillers, M. Isenburg, Isotropic surface remeshing, in: Proc. of the Shape Modeling Int., 2003.

[3] A. W. Bargteil, T. G. Goktekin, J. F. O'Brien, J. A. Strain, A semi-lagrangian contouring method for fluid simulation, ACM Trans. on Graphics 25 (1) (2006) 19–38.

[4] T. Brochu, Fluid animation with explicit surface meshes and boundary-only dynamics, Master's thesis, University of British Columbia (2006).

[5] D. Cohen-Steiner, P. Alliez, M. Desbrun, Variational shape approximation, in: SIGGRAPH, 2004.

[6] C. DeCoro, S. Rusinkiewicz, Pose-independent simplification of articulated meshes, in: Proc. of the Symp. on Interactive 3D graphics and games, 2005.

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

[8] M. A. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, M. B. Mineev-Weinstein, Roaming terrain: Real-time optimally adapting meshes, in: IEEE Visualization, 1997.

[9] D. Enright, F. Losasso, R. Fedkiw, A Fast and Accurate Semi-Lagrangian Particle Level-Set Method, Computers and Structures 83 (2005) 479–490.

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

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

[12] S. Goldenstein, C. Vogler, L. Velho, Adaptive Deformable Models for Graphics and Vision, Computer Graphics Forum 24 (4) (2005) 729–741.

[13] P. S. Heckbert, M. Garland, Optimal triangulation and quadric-based surface simplification, Comput. Geom. Theory Appl. 14 (1-3) (1999) 49–65.

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

[15] X. Jiao, Face offsetting: a unified framework for explicit moving interfaces, Journal of Computational Physics 220 (2) (2007) 612–625.

[16] X. Jiao, A. Colombi, X. Ni, J. C. Hart, Anisotropic Mesh Adaptation for Evolving Triangulated Surfaces, in: Proc. of 15th Int. Meshing Roundtable, 2006.

[17] S. Kircher, M. Garland, Progressive multiresolution meshes for deforming surfaces, in: ACM SIGGRAPH / Eurographics Symp. on Computer Animation, 2005.

[18] L. P. Kobbelt, T. Bareuther, H.-P. Seidel, Multiresolution shape deformations for meshes with dynamic vertex connectivity, Computer Graphics Forum 19 (3) (2000) 249–260.

[19] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, D. Dobkin, Maps: Multiresolution adaptive parameterization of surfaces, Computer Graphics Proc. (SIGGRAPH) (1998) 95–104.

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

[21] M. Meyer, M. Desbrun, P. Schroder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: VisMath, 2002.

[22] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer-Verlag, 2002.

[23] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1988) 12–49.

[24] A. Paiva, F. Petronetto, T. Lewiner, G. Tavares, Particle–based non–Newtonian fluid animation for melting objects, in: 19th Brazilian Symp. on Computer Graphics and Image Processing, Manaus, AM, 2006.

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

[26] P. Schröeder, Subdivision for modeling and animation, course notes of Siggraph. ACM SIGGRAPH. (1998).

[27] A. Shamir, C. L. Bajaj, V. Pascucci, Multi-resolution dynamic meshes with arbitrary deformations, in: IEEE Visualization, 2000.

[28] Y. Tong, P. Alliez, D. Cohen-Steiner, M. Desbrun, Designing quadrangulations with discrete harmonic forms, in: Proc. of Eurographics/ACM SIGGRAPH Symp. on Geometry processing, 2006.

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

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