# Meshless Thin-Shell Simulation Based on Global Conformal Parameterization

Xiaohu Guo, *Student Member*, *IEEE*, Xin Li, *Student Member*, *IEEE*,
Yunfan Bao, *Student Member*, *IEEE*, Xianfeng Gu, *Member*, *IEEE*, and Hong Qin, *Member*, *IEEE*

**Abstract**—This paper presents a new approach to the physically-based thin-shell simulation of point-sampled geometry via explicit, global conformal point-surface parameterization and meshless dynamics. The point-based global parameterization is founded upon the rigorous mathematics of Riemann surface theory and Hodge theory. The parameterization is globally conformal everywhere except for a minimum number of zero points. Within our parameterization framework, any well-sampled point surface is functionally equivalent to a manifold, enabling popular and powerful surface-based modeling and physically-based simulation tools to be readily adapted for point geometry processing and animation. In addition, we propose a meshless surface computational paradigm in which the partial differential equations (for dynamic physical simulation) can be applied and solved directly over point samples via Moving Least Squares (MLS) shape functions defined on the global parametric domain without explicit connectivity information. The global conformal parameterization provides a common domain to facilitate accurate meshless simulation and efficient discontinuity modeling for complex branching cracks. Through our experiments on thin-shell elastic deformation and fracture simulation, we demonstrate that our integrative method is very natural, and that it has great potential to further broaden the application scope of point-sampled geometry in graphics and relevant fields.

**Index Terms**—Meshless method, physically-based simulation, point-based geometry, surface parameterization, thin-shell.

## 1 Introduction

With the ever-increasing data acquisition power, point-sampled geometry is becoming ubiquitous in graphics and geometric information processing. One key reason for this new interest in points is that the polygonal complexity of graphical models has drastically increased in the last decade. In computer animation and physical simulation, complex physical effects, such as large deformation and cracks, pose grand technical challenges in terms of maintaining the topological consistency of the underlying meshes. The overhead of managing, processing, and manipulating very large polygonal-mesh connectivity information has given rise to the technical issue of the utility of polygons in many graphical applications, such as fracture animations, etc. For the simulation of complex physical phenomena, efficient and consistent surface representations are necessary to facilitate geometric and topological operations. For instance, in simulations of failure processes [42], [43], [44], we need to model the propagation of cracks along arbitrary and complex paths. This problem, in particular, becomes a notoriously difficult task using conventional mesh-based computational techniques such as finite element method (FEM), or finite difference method. In essence, the underlying structure of these methods, which stem from their reliance on meshes, impedes the flexible modeling and natural handling of discontinuities that do not coincide with the original mesh lines. Therefore, the most viable strategy for dealing with moving discontinuities in these methods is to remesh in each time step of the integration so that mesh lines remain coincident with the discontinuities throughout the simulation. However, this can introduce numerous difficulties for data management, such as the strong need to map between meshes in consecutive stages of the simulation, which inevitably results in degradation of both accuracy and complexity for system implementation. In addition, model remeshing becomes an unavoidable burden. To overcome the above difficulties associated with mesh structure in computer animation and simulation, a recent trend in this area is to utilize *meshless methods* [32] to model the physical behavior, which avoid complex remeshing operations and the associated problems of element cutting and mesh alignment sensitivity common in FEM [4].

In this paper, our task is to streamline the entire digital simulation and animation process (with specific application in thin-shell simulation) without any data conversion to meshes in order to further enhance the utility of point-sampled geometry and meshless methods and broaden its access by ordinary graphics users. Toward this goal, we propose to directly simulate the meshless dynamic behavior solely defined on global conformal parameterization of point surfaces.

In sharp contrast to our approach proposed in this paper, conventional techniques for modeling point-sampled surfaces are primarily based on implicit surface methods (Moving Least Squares [1] or Level Sets [8]) in recent years. One key motivation for the implicit representation is that point-sampled geometry has no explicit connectivity information. However, unlike popular polygonal meshes in graphics, the implicit surface representation is essentially a volumetric embedding. It does not admit a natural, two-dimensional domain for the effective analysis of point-sampled surfaces. As a result, simulating physical behavior

• *The authors are with the Center for Visual Computing and Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400. E-mail: {xguo, xinli, ybao, gu, qin}@cs.sunysb.edu.*

*Manuscript received 2 Aug. 2005; revised 3 Dec. 2005; accepted 16 Dec. 2005; published online 10 Mar. 2006.*
*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0099-0805.*

becomes rather difficult, compared with the direct processing of polygonal meshes. An easy and constructive way is to take a volumetric approach and to raise the analysis domain to three-dimensional space, where we perform the physical simulation in the 3D space enclosed by the surface, such as in [2]. Although the volumetric handling mechanism is both natural and intuitive, it is inappropriate to simulate thin-shell material. In many real-world applications, such as the deformation of the wings of the gargoyle (Fig. 10), or even open surfaces like a plate (Fig. 6), surface (thin-shell) simulation is much better than volumetric one, since one dimension of the surface body is much smaller than the other two, and the volumetric simulation may fail if the neighboring volumetric nodes are arranged in degenerate locations (such as a plane). To combat the deficiency associated with the dimensional increase, we resort to thin-shell simulation based on global parameterization as a more natural and efficient alternative.

The point surface parameterization technique in this paper is founded upon the rigorous mathematics of Riemann surface theory and Hodge theory. Parameterization is a fundamental and enabling tool for geometric analysis and synthesis. Most of the current parameterization techniques are only applicable to meshes with explicit connectivity information. These methods require converting the point cloud to mesh and then parameterizing the mesh. Intrinsically, the vicinity of points in the point cloud can provide enough information in order to unambiguously represent the topological information. The point cloud equipped with vicinity information is a much more simple structure compared with a mesh. In this paper, we assume that the point cloud is uniformly and well sampled, such that the surface construction is well defined locally within the neighborhood of the samples. Under such assumption, it is unnecessary to convert the point cloud to a mesh in order to explicitly represent the topology via connectivity. When doing so, general point cloud meshing methods most likely will lose some samples and decrease the accuracy and the quality of the geometric surface. A direct parameterization method for point clouds will be much simpler and more efficient without any loss of accuracy.

On the other hand, for physical simulations, typically the underlying formulations (from mathematical physics) require the solution of partial differential equations. Nowadays, meshless methods are becoming a powerful computational technique, especially for fracture simulations [35], [4]. The physics can be simulated on scattered sampling nodes using Moving Least Squares (MLS) approximation methods, which have been developed extensively in mechanical engineering and material science. The meshless methods have many desirable features, such as fast convergence, ease of adaptive refinement, flexible adjustment of the consistency order, and continuity of derivatives up to any desirable order, etc. The surface simulation nodes can be easily generated by applying quad-tree discretization of the parametric domain, whose subdivision depth depends on the conformal factor of the parameterization. The support size of each simulation node is also proportional to the conformal factor, which guarantees that their actual support size is consistent in $\mathbb{R}^3$. The quad-tree structure on the parametric domain can also be utilized to perform numerical integration via Quadrature, which is necessary in the meshless Galerkin weak form [33], [46] to ensure numerical accuracy and computational stability. Having the global parameterization, the crack branching problem is alleviated substantially and it is reduced to simple 2D line-intersection operations. Meanwhile, the fracture is modeled using the transparency criterion [36], and dynamic upsampling (node insertion) is applied in the vicinity of crack line to maintain numerical stability. All of the above operations can be easily performed on the 2D parametric domain. Essentially, the global conformal parameterization offers a common domain to facilitate accurate meshless simulation, and efficient discontinuity modeling for complex branching cracks.

**Contributions.** The specific and key contributions of this paper to the field of computer animation and graphics are:

1. We propose to simulate meshless thin-shell elastic deformation and crack propagation directly over point geometry, enabled by the global conformal parameterization. Compared with other local parameterization approaches, the global parameterization makes the physical simulation more accurate and stable.
2. At the geometric front, we systematically articulate a global conformal parameterization method for point-sampled surfaces.

## 2 Related Work

### 2.1 Physically-Based Animation

Pioneering work in the field of physically-based animation was originally carried out by Terzopoulos and his coworkers [25]. Thin-shell objects are naturally curved and can not be modeled using plate formulations [28]. Grinspun et al. [29] proposed a simple discrete shell model that can be derived geometrically for triangle meshes. For graphical modeling and animation of fracture process, most researchers rely on mesh-based methods. The works of [40], [41] simply break connections or springs between adjacent elements when the force exceeds a user-specified threshold value. The state of the art in fracture modeling for computer graphics is the works of [42], [43], which use a pseudo-principal stress and continuous remeshing. [44] proposed a virtual node algorithm that allows material to be separated along arbitrary piecewise-linear paths through a mesh. Pauly et al. [4] simulated volumetric meshless fracture with a highly dynamic surface and volume sampling method that affords complex fracture patterns of interacting and branching cracks.

### 2.2 Point-Based Geometry and Meshless Methods

Previous approaches for modeling and simulating point-sampled surfaces fall into two major categories: implicit and parametric methods. Alexa et al. [1] proposed to use the framework of moving least squares (MLS) projection to approximate a smooth surface defined by a set of points. Zwicker et al. [6] presented a system called Pointshop 3D for interactive shape and appearance editing of 3D point-sampled surfaces. Their patch correspondence is based on local minimum distortion parameterization. In [7], spherical

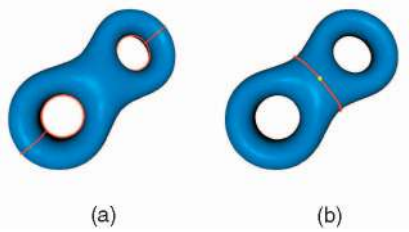


Fig. 1. (a) The canonical homology basis of a two-hole torus $\{a_1, b_1, a_2, b_2\}$, which are four closed curves. (b) The two-hole torus is cut into two patches by the handle separators (red line), which connect the two zero points (yellow dot).

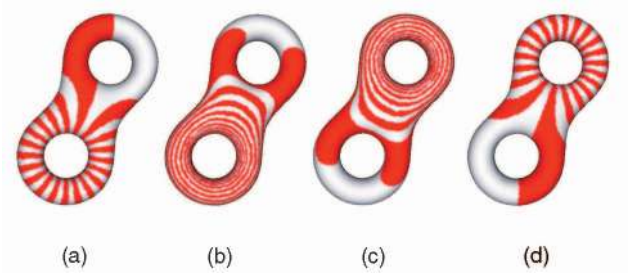


Fig. 2. The harmonic 1-forms $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ computed by (1). (a) $\omega_1$ dual to $a_1$, (b) $\omega_2$ dual to $b_1$, (c) $\omega_3$ dual to $a_2$, and (d) $\omega_4$ dual to $b_2$.

parameterization is proposed to parameterize genus zero point clouds.

Meshless methods were introduced into the graphics field by Desbrun and Cani in [26]. Later they applied Smoothed Particle Hydrodynamics (SPH) to simulate highly deformable bodies. Müller et al. [2] presented a method for modeling and animating elastic, plastic, and melting volumetric objects based on the MLS approximation of the displacement field. Their volumetric simulation does not require parameterization of the point-sampled surface. Most recently, they presented a geometrically motivated approach in [3] for simulating deformable point-based objects. Guo and Qin [45] combined meshless methods with modal analysis framework to provide real-time deformation of volumetric objects. The most relevant work to this paper is the approach proposed by Wicke et al. [5], which uses locally defined fibers to approximate the differential surface operators of the thin shell functional.

### 2.3 Conformal Parameterization

Several recent advances in surface parameterization [10] have been based on solving a discrete Laplace system [11], [12], [13], [14]. Lévy et al. [15] described a technique for finding conformal mappings by least squares minimization of *conformal energy*, and Desbrun et al. [17] formulated a theoretically equivalent method of discrete conformal parameterization. Sheffer and de Sturler [18] gave an angle-based flattening method for conformal parameterization.

Gu and Yau [19] considered construction of a global conformal structure for a manifold of arbitrary topology by finding a basis for holomorphic differential forms, based on Hodge theory [20]. Ni et al. [22] used the idea of a harmonic Morse function to extract the topological structure of a surface. Dong et al. [23] proposed a method for quadrilateral remeshing of manifolds using harmonic functions. The method is theoretically equivalent to using a holomorphic differential form as described in [19]. The differential forms in the latter, however, have at least four fewer zero points than those in the former.

## 3 Point-Based Parameterization

### 3.1 Theoretic Background

Global conformal parameterizations require some concepts and knowledge in both topology and Riemann surface. The algorithms for global conformal parameterizations are directly inspired by Riemann surface theory [30] and Hodge theory [20].

Essentially, finding a global conformal parameterization for point-set surfaces can be intuitively interpreted as the computation of a pair of smooth vector fields $(\omega_1, \omega_2)$ on the surface $S$, such that they satisfy the following criteria:

1. Both $\omega_1$ and $\omega_2$ have zero curl.
2. Both $\omega_1$ and $\omega_2$ have zero divergence.
3. $\omega_1$ and $\omega_2$ are conjugate to each other, namely $\omega_2$ can be obtained by rotating $\omega_1$ about the normal by a right angle counter-clockwisely everywhere on the surface. This operation is called Hodge star operation, denoted as $*$. Therefore, $\omega_2 = *\omega_1$.

Note that the above criteria are not independent. Conditions 1 and 3 can induce 2, and conditions 2 and 3 can induce 1. Vector fields $\omega_1, \omega_2$ with zero circulation and zero divergence are called *harmonic 1-forms*. The pair of conjugate harmonic 1-forms $(\omega_1, \omega_2)$ is called a *holomorphic 1-form*. The space of holomorphic 1-forms is isomorphic to the first cohomology group $H^1(S, R)$. Each cohomology class has a unique harmonic 1-form. Therefore, our goal is to find the harmonic 1-forms which compose a basis for $H^1(S, R)$.

From the knowledge of topology, we know that a genus $g$ closed surface has $g$ handles, and each handle has two special curves; one goes around the "hole," the other one circles the "tube." We denote such a pair of curves on handle $k$ as $a_k, b_k$; then, $a_k$ and $b_k$ intersect each other at one point, and $a_k, b_k$ do not intersect $a_j, b_j$ when $j \neq k$. Then, $2g$ curves $\{a_1, b_1, a_2, b_2, \cdots, a_g, b_g\}$ form a *canonical homology basis* (see Fig. 1a).

If we cut $S$ along $a_k$ to get an open surface $\tilde{S}$, then $\tilde{S}$ has two boundaries. We denote them as $a_k^+$ and $a_k^-$. If we define a *harmonic function* $f : \tilde{S} \to R$, such that $f|_{a_k^-} = 0$ and $f_{a_k^+} = 1$, and $f$ minimizes the *harmonic energy*,

$$E(f) = \int_{\tilde{S}} |\nabla f|^2,$$

then the gradient of $f$, $\nabla f$, is a harmonic 1-form on $S$. In this way, we can construct $2g$ harmonic 1-forms, denoted $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$ (see Fig. 2). They form the basis of the first cohomology group of $S$, $H^1(S, R)$. Then, pairs of vector fields $(\omega_1, *\omega_1), (\omega_2, *\omega_2), \cdots, (\omega_{2g}, *\omega_{2g})$ form a basis of all holomorphic 1-forms.

Given a holomorphic 1-form $\omega$, a *horizontal trajectory* is a curve on $S$ which is mapped to the horizontal curve (iso-$\theta_2$) on the parameter plane by integrating the holomorphic 1-form. A *vertical trajectory* is a curve on $S$ which is mapped to the iso-$\theta_1$ curve on the plane. The intersecting horizontal and vertical trajectories form the *conformal net*, which locally has a

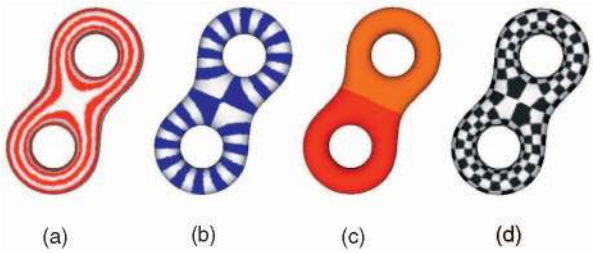


Fig. 3. (a) and (b) The parametric scalar fields (shown as isobars) in $\theta_1$ and $\theta_2$ directions, respectively, (c) the two-hole torus being separated into two patches by the handle separator, and (d) the global conformal parameterization visualized with a checkerboard texture.

tensor product structure. If $S$ is a closed Riemann surface with genus $g > 1$, the horizontal trajectories through the zero points of $\omega$ partition $S$ into cylinders (Fig. 1b), each of which can be conformally mapped to a rectangle by integrating $\omega$.

After we obtain a holomorphic 1-form $(\omega_1, \omega_2)$, we map the surface $S$ to the parametric plane by integration. Suppose that $(\theta_1, \theta_2)$ are their parametric values. First, we fix a base vertex $v_0$, for any vertex $v_k$, we select a curve on the surface $\gamma$ from $v_0$ to $v_k$, then we define the parameter of $v_k$ to be

$$(\theta_1(v_k), \theta_2(v_k)) = \int_\gamma (\omega_1, \omega_2).$$

Therefore, locally $\omega_1 = \nabla\theta_1$, $\omega_2 = \nabla\theta_2$. Because $\omega_1$ and $\omega_2$ are harmonic, the parameter $(\theta_1, \theta_2)$ of $v_k$ does not depend on the choice of $\gamma$, but depends on the homotopic class of $\gamma$. Fig. 3 shows the parametric scalar fields $(\theta_1, \theta_2)$ for the two-hole torus.

In order to improve the uniformity of the parameterizations, we need to introduce some boundaries to the surface. In the smooth case, because the parameterization is conformal, the metric of the surface can be written as $ds^2 = \lambda^2(\theta_1, \theta_2)(d\theta_1^2 + d\theta_2^2)$, where $(\theta_1, \theta_2)$ are the parameters, $\lambda$ measures the area distortion and is called the *conformal factor*. Different conformal parameterizations will severely affect the uniformity of $\lambda$. In general, we need to introduce boundaries to the tips of long tubes of the surface, such as the bunny's ear (Fig. 4). More detailed discussion can be found in [24].

Suppose the surface is of genus $g$ and has $b$ boundaries, then its doubling is closed, and with $2g + (b-1)$ genus. Therefore, the doubled surface has the cohomology group of $2(2g + (b-1))$ dimension. Because the doubled surface is symmetric, the original surface has the cohomology group of $2g + (b-1)$ dimension.

### 3.2 Algorithmic Overview

When computing the basis of the cohomology group, the conventional method [19] uses an algebraic topological approach, which depends on the combinatorial structure of the mesh. All connectivity information, boundary relation between faces, edges, and vertices have to be explicitly specified by the data structure. But, for the point cloud case, there is no such connectivity information. In contrast, it is easy to define functions on point clouds and compute their gradients. Since the topology of the surface is indicated by the singularities of Morse functions, we use Morse functions

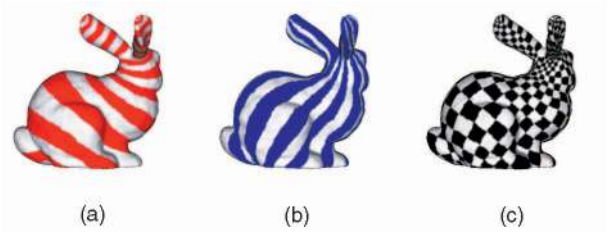


Fig. 4. The global conformal parameterization of the bunny model.

to compute the cohomology. If the surface has boundaries, the conventional method doubles the surface to make a symmetric closed surface. However, it is impossible to perform double covering for point clouds, since the point cloud can only represent a surface embedded in $\mathbb{R}^3$ and should be self-intersection free. In our point-based parameterization, we directly solve the diffusion equations corresponding to $b-1$ boundaries to get the $b-1$ harmonic 1-forms.

The algorithm flow to find the holomorphic 1-form basis can be summarized as follows:

1. Locate all boundaries of the surface $S$, and add them to a loop set $\Sigma$. Since we only need $b-1$ boundaries, we remove one boundary loop from $\Sigma$.
2. Compute the homology basis using fair Morse function [31], [22], then add all the loops to $\Sigma$.
3. For each loop $\tau$ from $\Sigma$, if $\tau \notin \partial S$, we cut the surface $S$ open along $\tau$ to get another surface $\tilde{S}$, and $\tau$ is split up to two loops $\tau^+, \tau^-$. Otherwise, we set $\tau^+ = \tau$ and set $\tau^-$ to be empty.
4. Compute a harmonic function $f : \tilde{S} \to R$, such that

$$\begin{aligned} \Delta f &= 0, \\ f|_{\tau^+} &= 1, \\ f|_{\gamma} &= 0, \quad \gamma \in \partial\tilde{S}, \quad \gamma \neq \tau^+, \end{aligned} \tag{1}$$

   where $\Delta$ is the Laplace-Beltrami operator on $\tilde{S}$. Note that the harmonic function $f$ equals 1 along $\tau^+$ and zero for all other boundary loops of $\tilde{S}$, including $\tau^-$.
5. Compute the gradient of $f$ on $\tilde{S}$, and translate $\nabla f$ to $S$. At each point on $S$, rotate $\nabla f$ about the normal a right angle to obtain another vector field $*\nabla f$; then, the pair of vector fields on $S$, $(\nabla f, *\nabla f)$ is a holomorphic 1-form corresponding to $\tau$.
6. Find all holomorphic 1-forms corresponding to all loops in $\Sigma$. These $2g + (b-1)$ holomorphic 1-forms compose a basis for all the holomorphic 1-forms on $S$.

Once we obtain the holomorphic 1-forms, we can find the map from the surface to the plane by integration. Because the topology of the surface is different from that of the plane, the map cannot be one to one everywhere. It can be shown that the number of singularities where the mapping is two to one is no more than the Euler number. The algorithm to parameterize a surface using a holomorphic 1-form is summarized as follows:

1. Choose a holomorphic 1-form, denoted as $\omega = (\omega_1, \omega_2)$, where $\omega_1$ and $\omega_2$ are vector fields. Locate their zero points.

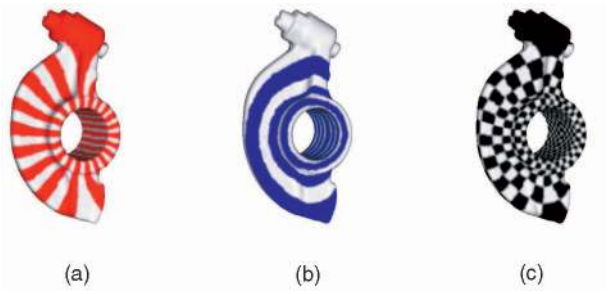


Fig. 5. The global conformal parameterization of the rocker arm model.

2. Trace the integral curve of $\omega_1$ through the zero points. Those curves segment the surface to several components, each component is a cylinder or a disk.
3. If a component is a cylinder, arbitrarily choose one point and trace the integral curve of $\omega_2$ to slice it open to a disk.
4. Integrate $(\omega_1, \omega_2)$ on each component and then map it to a rectangle.

The bunny model shown in Fig. 4 is a genus-0 surface. The user manually selects 3 points (2 at the ear tips, 1 at the bottom) as the opened boundary. We can compute two holomorphic 1-forms by repeating from step (3) to (5). After linear combination, we can get a desired holomorphic 1-form $(\omega_1, \omega_2)$. The integrated $(\theta_1, \theta_2)$ parameter value is shown in Fig. 4a and Fig. 4b. Fig. 5 shows the parameterization of the genus-1 rocker arm model.

### 3.3 Implementation

This section explains the implementation details of the above algorithms. We need to approximate the differential operators, such as the gradient, Laplace operator, the integration, and the Hodge star operator over point-set surfaces. The implementation in this paper is based on the assumption that all the input point clouds are locally uniformly-sampled surfaces, and the surface reconstruction is well defined within a neighborhood of the samples. For more theoretical results on the sampling conditions, please refer to [9].

**Approximating the gradient.** Suppose we choose one point $p$ in the point cloud $S$, and we set its neighborhood $N(p)$ as all the points inside a small sphere $|r - p| < \epsilon$; we denote a neighbor point as $q$. Given a function $f : S \to R$, the gradient $\nabla f(p)$ is approximated by minimizing the following energy:

$$E(\nabla f(p)) = \sum_q |f(q) - f(p) - \nabla f(p) \cdot (q - p)|^2.$$

The gradients are approximated in the tangent plane of $p$, i.e., $\mathbf{n} \cdot \nabla f = 0$. It can be also solved using the Moving Least Squares method [1], by associating each neighboring point with a distance-based weighting function.

**Approximating the Laplacian.** We use the *scale-dependent umbrella operator* [16] to approximate the Laplacian:

$$\Delta f(p) = \sum_{q_i \in N(p)} \varepsilon_{p,q_i}(f(q_i) - f(p)),$$

where the weights $\varepsilon_{p,q_i} = \frac{1}{\|q_i - p\|} / \sum_{q_j \in N(p)} \frac{1}{\|q_j - p\|}$ are in inverse proportion to their distances. We have also tried other Laplacian approximation methods, such as the approaches using the shape-preserving weights [13], or mean value weights [14]. In our experiments, we found no significant visual difference between the more complicated Laplacian approximation schemes and the simpler reciprocal distance scheme. Therefore, we choose this simplest formulation in our implementation.

Any function defined on $S$ can be deformed to be a harmonic function using the following flow method in [22], [12],

$$\frac{\partial f}{\partial t} = -\Delta f.$$

In addition, harmonic functions have the minimal number of extrema.

**Locating extremal points.** At the extremal point $p$, the gradient $\nabla f(p)$ is extremely close to zero. We first estimate the gradient, then find the cluster of points with relatively small gradient lengths and compute the center of gravity of each cluster. We pick the point on $S$ which is closest to it as the extremal point. For a genus $g$ closed surface, there are generally $2g$ extrema other than the maxima $p_1$ and minima $p_0$.

**Tracing integral curves.** Tracing the integral curve along $\nabla f$ is straightforward. At the current point $p$ with gradient $\nabla f(p)$, we add a new traced point in the direction of $\nabla f(p)$ on its tangent plane at a controllable distance to $p$. Then, we can project the traced point to the original point-sampled surface by the MLS projection operator [1].

**Computing homology basis using fair Morse function.** In order to compute the homology basis, we use a method based on Morse theory. Essentially, we define a function on the surface, and then we make the function as smooth as possible to reduce the number of extremal points. The integral curve along the gradient of the function gives the homology basis. Details can be found in [31], [22]. Since we use least squares to fit a gradient vector for each point considering all neighboring information, it is more accurate than purely using piecewise linear approximation on a mesh. If we can estimate the gradient accurately, we can find critical points and trace homology bases robustly.

**Approximating the Hodge star operation.** The Hodge star operator on a vector field is simply to rotate the vector about the normal $90^\circ$ at every point on the surface,

$$* \nabla f(p) = \mathbf{n}(p) \times \nabla f(p).$$

**Approximating integration.** In order to parameterize a surface patch $\Gamma \subset S$, we integrate a holomorphic 1-form $\omega = (\omega_1, \omega_2)$. Since the integration process is path independent theoretically, we can design it as follows: First, we fix a base point $p \in \Gamma$; for any point $q \in \Gamma$, we choose a sequence of points from $p$ to $q$ as the path $r$, $r = \{p_0, p_1, p_2, \cdots, p_n\}$, where $p_0 = p$, $p_n = q$, then

$$(\theta_1, \theta_2) = \sum_{i=1}^{n} (\omega_1(p_{i-1}) \cdot (p_i - p_{i-1}), \omega_2(p_{i-1}) \cdot (p_i - p_{i-1})).$$

## 4 Meshless Thin-Shell Simulation

For any point-sampled surface, if we assume that one dimension (i.e., the thickness), of the surface body is

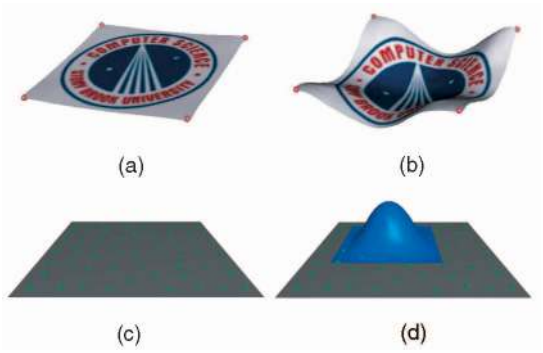


Fig. 6. Point-based thin-shell before (a) and after (b) deformations, with the simulation nodes scattered on the parameter plane (c). The shape function associated with one of the nodes is shown in (d).

significantly smaller than the other two dimensions, we can consider the point-sampled surface as a thin-shell. In the Kirchhoff-Love thin shell framework, the deformation of the surface body is fully described by the deformation of the middle surface represented by point-samples. Let $\varphi$ denote the position of a point on the middle surface of the shell, and let $\mathbf{a}_3$ be the unit director vector which is normal to the shell surface. Given the global parameterization of the point-sampled surfaces, we can describe the positions of any material point in the reference (denoted $\bar{\mathbf{r}}$) and deformed (denoted $\mathbf{r}$) configurations of the shell by:

$$\bar{\mathbf{r}}(\theta_1,\theta_2,\theta_3) = \bar{\varphi}(\theta_1,\theta_2) + \theta_3\bar{\mathbf{a}}_3(\theta_1,\theta_2), \tag{2}$$

$$\mathbf{r}(\theta_1,\theta_2,\theta_3) = \varphi(\theta_1,\theta_2) + \theta_3\mathbf{a}_3(\theta_1,\theta_2), \tag{3}$$

where $\theta_1$ and $\theta_2$ are parameters of the point-sampled middle surface, and $\theta_3$ ($-\frac{h}{2} \le \theta_3 \le \frac{h}{2}$) is in the thickness direction.

The precise form of the membrane and bending strain and stress matrices are given in [28]. We use the *Euler-Lagrange* equations for our elastic deformation:

$$\frac{d}{dt}\left(\frac{\partial T(\dot{\mathbf{u}})}{\partial \dot{\mathbf{u}}}\right) + \mu\dot{\mathbf{u}} + \frac{\partial V(\mathbf{u})}{\partial \mathbf{u}} = \mathbf{F}_{ext}, \tag{4}$$

where the kinetic energy $T$ and elastic potential energy $V$ are functions of $\dot{\mathbf{u}}$ and $\mathbf{u}$, respectively. The term $\mu\dot{\mathbf{u}}$ is the generalized damping force, and $\mathbf{F}_{ext}$ is a generalized force arising from users' applied forces, or external body forces, such as gravity. The kinetic energy of the moving body can be expressed as:

$$T = \frac{1}{2}\int_\Omega h\rho(\mathbf{x})\dot{\mathbf{u}}\cdot\dot{\mathbf{u}}d\Omega = \frac{1}{2}\sum_{I,J}\mathbf{M}^{IJ}\dot{\mathbf{u}}_I\cdot\dot{\mathbf{u}}_J, \tag{5}$$

where $\rho(\mathbf{x})$ is the mass density of the body, and $\mathbf{M}^{IJ} = \int_\Omega h\rho(\mathbf{x})\phi_I(\mathbf{x})\phi_J(\mathbf{x})d\Omega$. The elastic potential energy $V$ is given by the formula:

$$V = \int_\Omega [\frac{Eh}{1-\nu^2}\alpha^T\tilde{\mathbf{H}}\alpha + \frac{Eh^3}{12(1-\nu^2)}\beta^T\tilde{\mathbf{H}}\beta]d\Omega, \tag{6}$$

where $\alpha$ and $\beta$ are the membrane and bending strains, respectively, $\tilde{\mathbf{H}}$ is the standard constitutive matrix, the constant $E$ is *Young's modulus*, and the coefficient $\nu$ is *Poisson's ratio*. More specific derivations can be found in [28].

We utilize the meshless methods to discretize the kinetic energy $T$ and the elastic potential energy $V$. The advantages of the meshless methods may be summarized as follows:

1. they can easily handle very large deformations, since the connectivity among nodes is generated as part of the computation and can change over time;
2. moving discontinuities such as cracks can be naturally facilitated, since no new mesh needs to be constructed as in finite element methods, and the computational cost of remeshing at each time step can be avoided;
3. accuracy can be controlled more easily, since in areas where more refinement is needed, nodes can be added quite flexibly;
4. data management overhead can be minimized during simulation.

### 4.1 Moving Least Squares Shape Functions

In this paper, the shape functions are constructed by using the MLS technique [33] or, alternatively, on the basis of reproducibility conditions (note that both approaches can arrive at the same expressions for the shape functions). The MLS method can provide continuous and smooth field approximation throughout the analysis domain with any desirable order of consistency. We associate each node $I$ with a positive weight function $w_I$ of compact support. The support of the weight function $w_I$ defines the domain of influence of the node: $\Omega_I = \{\mathbf{x}\in\mathbb{R}^2 : w_I(\mathbf{x}) = w(\mathbf{x},\mathbf{x}_I) > 0\}$, where $w(\mathbf{x},\mathbf{x}_I)$ is the weight function associated with node $I$ evaluated at the parametric position $\mathbf{x} = (\theta_1,\theta_2)$. The approximation of the field function $f$ at $\mathbf{x}$ is only affected by those nodes whose weights are nonzero at $\mathbf{x}$. We call the set of such nodes the *active set* $\mathcal{A}(\mathbf{x})$. In the thin-shell simulation, if we consider the displacement field as a function of both space and time $\mathbf{u}(\mathbf{x},t)$, the approximation in the parametric domain $\Omega$ can be written as:

$$\mathbf{u}(\mathbf{x},t) \approx \mathbf{u}^h(\mathbf{x},t) = \sum_{I\in\mathcal{A}(\mathbf{x})}\phi_I(\mathbf{x})\mathbf{u}_I(t), \tag{7}$$

where $\phi_I(\mathbf{x})$ is called the shape function of the MLS approximation, and $\mathbf{u}_I(t)$ is the nodal displacement value. To obtain consistency of any desirable order of approximation, it is necessary to have a complete basis. For a complete polynomial basis of order $n$, $\mathbf{p}(\mathbf{x}) = [1 \quad \mathbf{x}\ldots\mathbf{x}^n]^T$, the shape function can be derived as:

$$\phi_I(\mathbf{x}) = w_I(\mathbf{x})\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_I), \tag{8}$$

where $\mathbf{A}(\mathbf{x})$ is called the *moment matrix*. Due to the page limitation, we refer the readers to [33] for more detailed derivation.

In the thin-shell simulation, positional constraints can be achieved by adding constraint forces as Lagrange multipliers [38]. Fig. 7 shows the meshless shell deformation on the bunny models of different stiffness material by user's applied forces. The user enforces certain positional constraints on the bottom of the bunny (small pink balls), and applied two constant forces (blue arrows) at the end of the two ears.

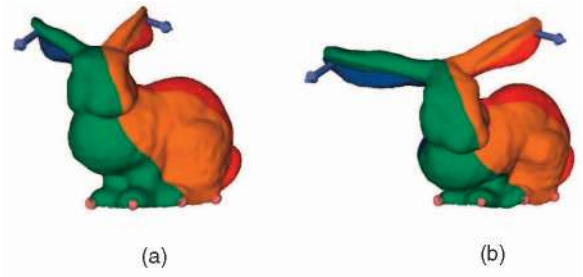


Fig. 7. (a) Deformation of the bunny with *Young's modulus* = $5 \times 10^7$. (b) *Young's modulus* = $1 \times 10^7$.

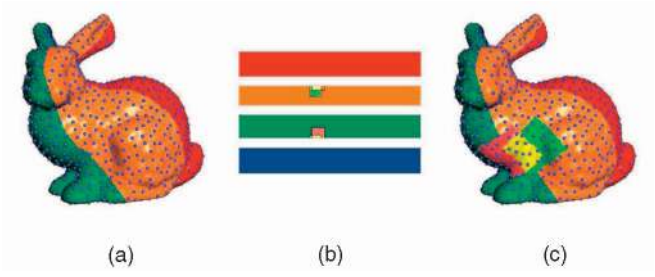


Fig. 8. (a) Uniform sampling nodes on the surface. (b) The four parametric planes. (c) The supporting regions of two simulation nodes are shown in pink and green; their overlapping region is shown in yellow.

### 4.2 Sampling on the Parametric Domain

One important advantage of meshless methods is the flexibility of the sampling pattern. In the thin-shell simulation, it would be more desirable to have an initial sampling scheme such that the sampling nodes are uniformly distributed on the manifold surface (Fig. 8a). Similar to the idea of using an octree structure to facilitate the volumetric sampling of the volumetric space [37], we utilize a quad-tree structure on the parametric domain. The subdivision depth of the quad-tree is dependent on the conformal factor $\lambda$. Suppose the size of the quad-tree cell is $l$. If $\lambda l$ is larger than a threshold (i.e., the surface patch corresponding to the cell is still large enough), we keep subdividing the cell into four child-cells. We place the sampling nodes based on the quad-tree discretization of the parametric plane. Since the conformal factors in $u$ and $v$ directions are equivalent, we can choose a simple "symmetric" supporting region (such as squares, or disks) for each simulation nodes. In our implementation, we use a square-shaped supporting region to ease numerical integrations. For a node $i$ residing in quad-tree cell $Q_i$, its supporting region is a square of size $\eta \cdot size(Q_i)$ centered around node $i$. We require the quad-tree to be a one-level adjusted quad-tree, where the level difference of all terminal cells and their edge neighbors is no more than one. This restriction can guarantee the automatic satisfaction of the patch covering condition in order to make the moment matrix invertible. It has been proved in [37] that by choosing a suitable size for $\eta$, the validity of the supporting region can be guaranteed a priori. For example, for a linear basis $\mathbf{p}(\mathbf{x})^T = [1, \theta_1, \theta_2]$, any point in the domain will be covered by at least three patches if we choose $\eta$ to be 3. The supporting region construction based on terminal quad-tree cells can provide the structure needed to perform efficient neighboring search and patch intersection tests. The quad-tree cells can be also utilized as integration cells to perform numerical integration for (5) and (6). Note that using $\lambda$ to control the subdivision depth of the quad-tree is just an approximating approach to achieve uniform distribution of the nodes. In some other applications (such as adaptive simulation), it may be desirable to place more simulation nodes in areas where more simulation details are needed, in which cases new control coefficients (such as material properties, etc.) can be introduced to control the subdivision depth.

In the parameterization stage, the seams between different parametric patches are represented by additional point set curves. They are generated by tracing integral curves (Section 3.3). These additional point set curves are only used in the parameterization step, i.e., they are not added to the original point set surface after the parameterization. In the simulation stage, the parametric seams are maintained in a table indicating the connection correspondence of different parametric patches. The simulation nodes are not duplicated on the patch boundaries. The support of the shape function associated with each simulation node is not restricted to the parametric patch where it resides. In fact, the support can be extended to another parametric patch if the node is close to the boundary of its patch (see Fig. 8c). So, the behavior of the nodes across the boundaries is consistent with nonboundary nodes without any unnatural artifacts.

### 4.3 Modeling Cracks

Typically, there are two aspects of crack simulations that are of interest: 1) The physical model undergoing the crack evolution. We use the simplified condition of maximal principle stress [42], [44] to decide both where and how the material cracks. If the maximum eigenvalue of the stress exceeds a threshold, a crack line (with cracking speed proportional to the maximum eigenvalue of the stress) should be generated. Similar to [44], secondary fractures on the cracking surface can be given higher thresholds to help reduce spurious branching in practice. 2) The representation of the evolving geometry. For thin-shell crack simulation, the evolving geometry can be simply represented as line segments on the 2D parametric domain.

When a crack is generated in a body, the dependent variables (e.g., the displacements, etc.), must be discontinuous across the crack. Furthermore, the support of the nodes affected by the discontinuities need to be modified accordingly to incorporate the proper behavior of the shape functions and its derivatives. The simplest way to introduce discontinuities into meshless approximations is to use the visibility criterion [32]. However, it would cause undesirable discontinuities of the shape functions within the domain. Similar to the approach of [4], we use the transparency criterion proposed in [36] to allow partial interaction of nodes in the vicinity of the crack front. Note that all these operations can be easily performed on the 2D parametric domain.

We need to perform dynamic up-sampling (node insertion) during the crack simulation, in order to maintain numerical stability even for an initially adequately sampled model. New simulation nodes need to be inserted in the vicinity of crack lines. We take the same criterion as [4] to determine under-sampling at each node based on transparency weights. If the transparency weights become too small

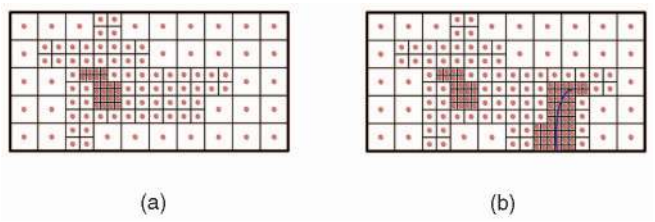


Fig. 9. (a) The quad-tree structure on the parametric plane for placing simulation nodes. (b) The dynamic resampling near the crack line (blue curve).

due to a nearby crack line, we subdivide the quad-tree cell associated with the simulation node into 4 child nodes (see Fig. 9b).

### 4.4 Global Conformal Parameterization versus Local Parameterization

For point-based surfaces, although it seems to be possible to perform the meshless simulation based on local parameterizations, such as [39], [5], the numerical accuracy and stability would be hard to enforce.

In [39], the neighboring points are projected onto the local tangent plane and local triangulations are utilized to assemble the FEM matrices. However, using the local tangent plane as local parameterization domain will decrease the simulation accuracy especially for high curvature areas, since the surface metric may not be preserved accurately on the tangent plane and it would not be valid to choose simple and commonly-used supporting region (such as squares or disks) for the simulation nodes. Although it is possible to choose other local parameterizations instead of local tangent planes, enforcing consistencies among neighboring local parameterizations would become computationally intractable. For example, it would be very difficult to perform the integration for the mass and stiffness matrices under the meshless setting, since the integration domain $\Omega$ in (5) and (6) is not unique or consistent in each local parameterization domain. So, in [34], they have to resort to a subdivision mesh as a background mesh for numerical integration.

Wicke et al. [5] proposed to use locally defined fibers to approximate the differential surface operators of the thin shell functional. Their fracture was modeled by cutting fibers between simulation nodes. However, the continuity and consistency requirements are extremely hard to guarantee in their approach, especially in the situation of complex branching cracks. In this paper, we use the MLS approximant which can easily satisfy any consistency order and continuity requirement. The fracture is modeled using the transparency criterion, and dynamic upsampling (node insertion) in the vicinity of a crack line is applied to maintain numerical stability. All these operations can be made easy by providing a global conformal parameterization of the original point-sampled surfaces, since they all operate on the 2D parametric domain and crack branching becomes simple 2D line-intersection operation.

## 5 Experimental Results

Our system is implemented on a Microsoft Windows XP PC with dual Intel Xeon 2.0GHz CPUs, 1.5GB RAM, and an nVidia GeForce Fx 5200 Ultra GPU. The entire point-based rendering pipeline is built upon Pointshop 3D [6]. Table 1 shows the statistics and performance data of our experiments.

The computational consumption in the parameterization stage consists of the timing of the entire algorithmic flow specified in Section 3.2 and Section 3.3, such as finding all the boundaries, solving fair Morse functions to get the homology basis, solving Laplacian equations for the harmonic functions, performing integrations, etc. They also include all the system response time (e.g., CPU computation, memory access, etc.). Note that for high genus models, we solve $2g + (b - 1)$ Laplacian equations corresponding to different homology bases. Similar to [13], we use the Bi-CGSTAB iterative method to solve the sparse linear systems. The global conformal parameterization time for most of the models we experimented with takes around two minutes. The largest model in our experiments is the Iphigenie model (200,219 points, Fig. 12), which can be parameterized in 15 minutes. The global parameterization of the point-sampled surfaces are computed offline, while the mesh-free simulation takes the parameterization results as input.

In the simulation stage, we use the implicit time integration method, which has been demonstrated to be very stable. The initial numbers of nodes in the examples of gargoyle, bunny, and Iphigenie simulations are only around 500. The elastic deformation of the gargoyle (Fig. 10) and bunny (Fig. 7) is simulated in real-time. When deforming the wings of the gargoyle model, the user specified several positional constraints at the head and bottom of the model, and applied forces to bend the wings. Note that the forces are not constant over the deformation. In fact, the user dynamically changes the force magnitude and direction, since the more bending effect the wings achieve, the larger forces the user needs to apply. The computational load at

TABLE 1
Model Statistics and Performance Data

| Example | Figure | Points | Parameterization (sec) | Nodes (Initial/Final) | Simulation (sec/frame) |
|---|---|---|---|---|---|
| two-hole torus | 3 | 29,998 | 133.89 | N/A | N/A |
| rocker arm | 5 | 40,000 | 88.63 | N/A | N/A |
| gargoyle (deform.) | 10 | 40,002 | 62.09 | 520/520 | 0.026 |
| bunny (deform.) | 4 & 7 | 50,002 | 99.85 | 440/440 | 0.022 |
| bunny (fracture) | 4 & 11 | 50,002 | 99.85 | 440/485 | 1.3 |
| Iphigenie (fracture) | 12 | 200,219 | 866.45 | 420/650 | 1.2 |

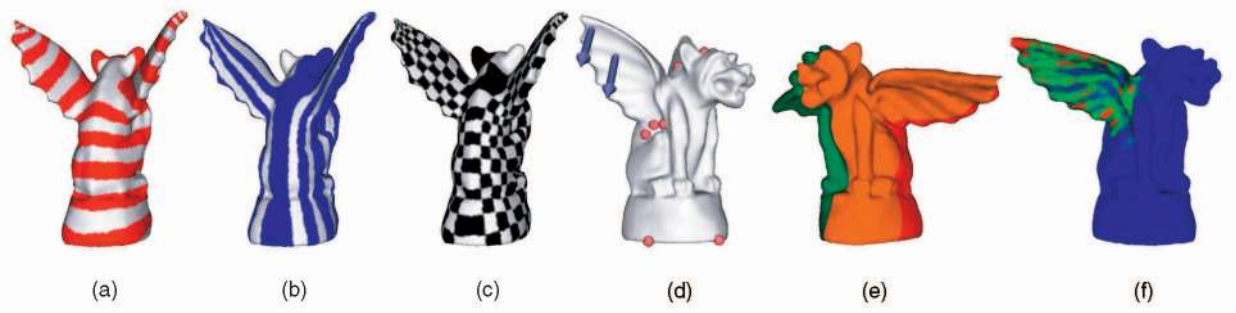


Fig. 10. (a), (b), and (c) The global conformal parameterization of the gargoyle model. (d) User's applied forces and specified position constraints. (e) The deformation of the gargoyle's wings (different parametric patches are shown in different colors). (f) The stress distribution after deformation.

each time step of the crack propagation is much higher than pure elastic deformation. We need to perform transparency tests for the integration points, surface points, and simulation nodes in the neighborhood of new crack lines. And, new simulation nodes need to be generated if necessary. We have to update the mass and stiffness matrices at each time step to accomodate these changes. Fig. 11 shows an example of tearing the bunny model. The user sets a positional constraint at one ear tip of the bunny, while applying the force at another ear tip to tear the whole bunny open. Fig. 12 shows the global conformal parameterization and explosion simulation[1] of the Iphigenie model. The crack simulations of the bunny and Iphigenie can be performed at around 1.2 sec/frame, while the overall simulations take around 10 minutes to complete.

## 6 Conclusion

We have presented a meshless thin-shell simulation framework solely based on global conformal parameterization of the point-sampled surfaces. The parameterization is founded upon Riemann surface theory and Hodge theory. The point cloud is a valid geometric representation of surfaces. Therefore, the conformal structure of surfaces can be derived from this simple representation. This paper is the first attempt to tackle this fundamental problem. The structure of the linear solution space of all global conformal parameterizations on point samples is uniquely determined by the manifold geometry and independent of the connectivity of the surface points. The vicinity information of these point samples provides enough information to unambiguously represent the intrinsic global topological information. The global parameterization is a fundamental process for the meshless thin-shell simulation, especially for the fracture simulations. Both the parameterization and dynamic simulation processes are only built upon point-samples without any connectivity information. Because of the structural simplicity of the underlying representation, our new algorithms are efficient and easy to implement.

There are many avenues for near-future work. First, we only assume that the point surface is sufficiently and regularly sampled. The sampling issue associated with point geometry is far from trivial, and it requires tremendous new research efforts in the near future to do a complete justice to their quantitative influences on the parameterization quality. Second, our global parameterization work is expected to pave the way for our ongoing research and future projects in point-based graphics. Looking beyond thin-shell simulations and our own research, the point-surface global parameterization framework developed in this paper can readily facilitate other research initiatives in many graphics applications, such as shape registration/analysis, segmentation, cut-and-paste, morphing, attribute transfer, texture synthesis, spline surface fitting, etc. It has a great potential to maximize the utility of point-sampled surfaces, while simultaneously retaining their structural simplicity. With the growing demand on effectively processing topologically complicated point geometries in 3D, our techniques are poised to contribute to the state-of-the-art of graphics and interactive techniques.

1. This experiment is only for the purpose of scientific simulation/animation, with no offense to any religious or culture significance of Iphigenie.

## Acknowledgments

The authors thank the Stanford Computer Graphics Laboratory for the use of the bunny model. The rocker arm model is courtesy of Cyberware Inc. They appreciate and thank the team at ETH Zurich who developed and published Pointshop 3D. This research was supported in part by US National Science Foundation grants IIS-0082035 and IIS-0097646, and an Alfred P. Sloan Fellowship.

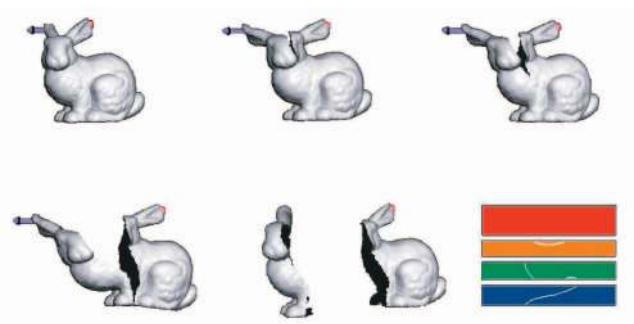

Fig. 11. Tearing the bunny model by the user's applied force. The lower right figure shows the crack lines in the parametric domain.

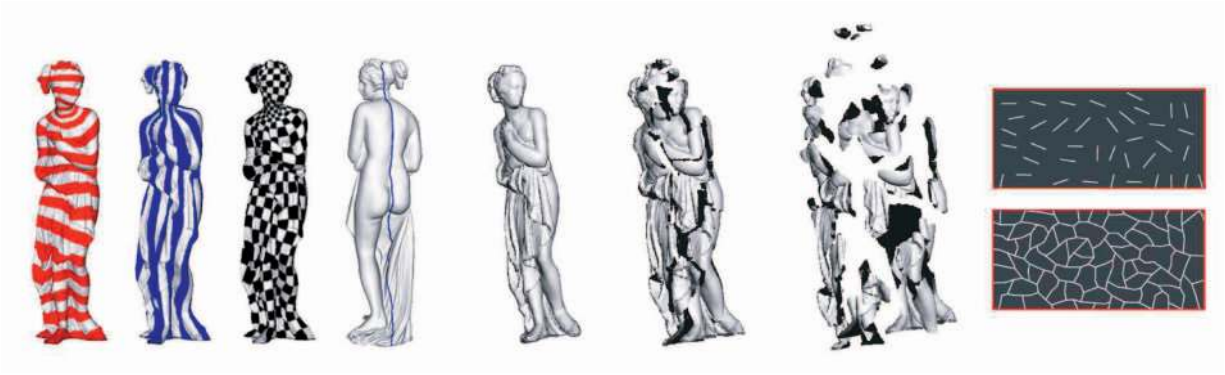

Fig. 12. Global conformal parameterization and meshless simulation of the explosion of Iphigenie. The fourth figure shows the traced integral curve connecting two boundaries in the parameterization step. The right most figure shows the crack pattern of the parametric domain in the first (top) and final (bottom) step of the simulation.

## REFERENCES


[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva, "Computing and Rendering Point Set Surfaces," *IEEE Trans. Visualization and Computer Graphics,* vol. 9, no. 1, pp. 3-15, Jan.-Mar. 2003.

[2] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa, "Point Based Animation of Elastic, Plastic and Melting Objects," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation,* pp. 141-151, 2004.

[3] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, "Meshless Deformations Based on Shape Matching," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 471-478, 2005.

[4] M. Pauly, R. Keiser, B. Adams, P. Dutré, M. Gross, and L.J. Guibas, "Meshless Animation of Fracturing Solids," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 957-964, 2005.

[5] M. Wicke, D. Steinemann, and M. Gross, "Efficient Animation of Point-Sampled Thin Shells," *Proc. Computer Graphics Forum,* vol. 24, no. 3, pp. 667-676, 2005.

[6] M. Zwicker, M. Pauly, O. Knoll, and M. Gross, "Pointshop3D: An Interactive System for Point-Based Surface Editing," *ACM Trans. Graphics,* vol. 21, no. 3, pp. 322-329, 2002.

[7] M. Zwicker and C. Gotsman, "Meshing Point Clouds Using Spherical Parameterization," *Proc. Eurographics Symp. Point-Based Graphics,* 2004.

[8] X. Guo, J. Hua, and H. Qin, "Scalar-Function-Driven Editing on Point Set Surfaces," *IEEE Computer Graphics and Applications,* vol. 24, no. 4, pp. 43-52, July-Aug. 2004.

[9] P.T. Bremer and J.C. Hart, "A Sampling Theorem for MLS Surfaces," *Proc. Eurographics Symp. Point-Based Graphics,* pp. 47-54, 2005.

[10] M.S. Floater and K. Hormann, "Surface Parameterization: A Tutorial and Survey," *Advances in Multiresolution for Geometric Modelling,* pp. 157-186, 2005.

[11] U. Pinkall and K. Polthier, "Computing Discrete Minimal Surfaces and Their Conjugates," *Experimental Math.,* vol. 2, no. 1, pp. 15-36, 1993.

[12] T. Duchamp, A. Certain, A. DeRose, and W. Stuetzle, "Hierachical Computation of PL Harmonic Embeddings," technical report, Univ. of Washington, July 1997.

[13] M.S. Floater, "Parametrization and Smooth Approximation of Surface Triangulations," *Computer Aided Geometric Design,* vol. 14, no. 3, pp. 231-250, 1997.

[14] M.S. Floater, "Mean Value Coordinates," *Computer Aided Geometric Design,* vol. 20, no. 1, pp. 19-27, 2003.

[15] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least Squares Conformal Maps for Automatic Texture Atlas Generation," *ACM Trans. Graphics,* vol. 21, no. 3, pp. 362-371, 2002.

[16] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr, "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow," *Proc. SIGGRAPH,* pp. 317-324, 1999.

[17] M. Desbrun, M. Meyer, and P. Alliez, "Intrinsic Parameterizations of Surface Meshes," *Proc. Eurographics,* pp. 209-218, 2002.

[18] A. Sheffer and E. de Sturler, "Parameterization of Faceted Surfaces for Meshing Using Angle-Based Flattening," *Eng. Computations,* vol. 17, no. 3, pp. 326-337, 2001.

[19] X. Gu and S.T. Yau, "Global Conformal Surface Parameterization," *Proc. Eurographics/SIGGRAPH Symp. Geometry Processing,* pp. 127-137, 2003.

[20] R. Schoen and S.T. Yau, *Lectures on Harmonic Maps.* Int'l Press, 1997.

[21] C.L. Siegel, *Lectures on Quadratic Forms,* Tata Inst. of Fundamental Research, 1957.

[22] X. Ni, M. John, and J. Hart, "Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 613-622, 2004.

[23] S. Dong, S. Kircher, and M. Garland, "Harmonic Functions for Quadrilateral Remeshing of Arbitrary Manifolds," *Computer Aided Geometric Design,* vol. 22, no. 5, pp. 392-423, 2005.

[24] M. Jin, Y. Wang, S.T. Yau, and X. Gu, "Optimal Global Conformal Surface Parameterization," *Proc. Visualization Conf.,* pp. 267-274, 2004.

[25] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically Deformable Models," *Proc. SIGGRAPH,* pp. 205-214, 1987.

[26] M. Desbrun and M.P. Cani, "Animating Soft Substances with Implicit Surfaces," *Proc. SIGGRAPH,* pp. 287-290, 1995.

[27] M. Desbrun and M.P. Cani, "Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies," *Proc. Eurographics Workshop Animation and Simulation,* pp. 61-76, 1996.

[28] F. Cirak, M. Ortiz, and P. Schröder, "Subdivision Surfaces: A New Paradigm for Thin-Shell Finite Element Analysis," *Int'l J. Numerical Methods in Eng.,* vol. 47, no. 12, pp. 2039-2072, 2000.

[29] E. Grinspun, A.N. Hirani, M. Desbrun, and P. Schröder, "Discrete Shells," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation,* pp. 62-67, 2003.

[30] J. Jost, *Compact Riemann Surfaces.* Springer, 2000.

[31] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder, "Removing Excess Topology from Isosurfaces," *ACM Trans. Graphics,* vol. 23, no. 2, pp. 190-208, 2004.

[32] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless Methods: An Overview and Recent Developments," *Computer Methods in Applied Mechanics and Eng.,* vol. 139, pp. 3-47, 1996.

[33] T. Belytschko, Y.Y. Lu, and L. Gu, "Element Free Galerkin Methods," *Int'l J. Numerical Methods in Eng.,* vol. 37, pp. 229-256, 1994.

[34] P. Krysl and T. Belytschko, "Analysis of Thin Shells by the Element-Free Galerkin Method," *Int'l J. Solids and Structures,* vol. 33, pp. 3057-3080, 1996.

[35] P. Krysl and T. Belytschko, "The Element Free Galerkin Method for Dynamic Propagation of Arbitrary 3-D Cracks," *Int'l J. Numerical Methods in Eng.,* vol. 44, pp. 767-800, 1999.

[36] D. Organ, M. Fleming, T. Terry, and T. Belytschko, "Continuous Meshless Approximations for Nonconvex Bodies by Diffraction and Transparency," *Computational Mechanics,* vol. 18, pp. 1-11, 1996.

[37] O. Klaas and M.S. Shephard, "Automatic Generation of Octree-Based Three-Dimensional Discretizations for Partition of Unity Methods," *Computational Mechanics,* vol. 25, pp. 296-304, 2000.

[38] D. Metaxas and D. Terzopoulos, "Dynamic Deformation of Solid Primitives with Constraints," *Proc. SIGGRAPH,* pp. 309-312, 1992.

[39] U. Clarenz, M. Rumpf, and A. Telea, "Finite Elements on Point Based Surfaces," *Proc. Eurographics Symp. Point-Based Graphics,* 2004.
[40] K. Hirota, Y. Tanoue, and T. Kaneko, "Generation of Crack Patterns with a Physical Model," *The Visual Computer,* vol. 14, no. 3, pp. 126-137, 1998.
[41] J. Smith, A. Witkin, and D. Baraff, "Fast and Controllable Simulation of the Shattering of Brittle Objects," *Computer Graphics Forum,* vol. 20, no. 2, pp. 81-90, 2001.
[42] J.F. O'Brien and J.K. Hodgins, "Graphical Modeling and Animation of Brittle Fracture," *Proc. SIGGRAPH,* pp. 137-146, 1999.
[43] J.F. O'Brien, A.W. Bargteil, and J.K. Hodgins, "Graphical Modeling and Animation of Ductile Fracture," *Proc. SIGGRAPH,* pp. 291-294, 2002.
[44] N. Molino, Z. Bao, and R. Fedkiw, "A Virtual Node Algorithm for Changing Mesh Topology during Simulation," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 385-392, 2004.
[45] X. Guo and H. Qin, "Real-Time Meshless Deformation," *Computer Animation and Virtual Worlds,* vol. 16, nos. 3-4, pp. 189-200, 2005.
[46] G. Lecot, B. Lévy, L. Alonso, and J.C. Paul, "Master-Element Vector Irradiance for Large Tessellated Models," *Proc. Graphite,* pp. 315-322, 2005.

**Xiaohu Guo** received the BS degree (2001) in computer science from the University of Science and Technology of China. He received the MS degree (2004) in computer science from the State University of New York at Stony Brook. He is a PhD candidate in the Department of Computer Science at the State University of New York at Stony Brook. His research interests include computer graphics, geometric and physics-based modeling, computer animation and simulation, scientific visualization, human-computer interaction, virtual reality, and computer vision. For more information, please visit http://www.cs.sunysb.edu/~xguo. He is a student member of the IEEE and the IEEE Computer Society.

**Xin Li** received the BS degree (2003) in computer science from the University of Science and Technology of China. He received the MS degree (2005) in computer science from the State University of New York at Stony Brook. He is a PhD candidate in the Department of Computer Science at the State University of New York at Stony Brook. His research interests include computer graphics, geometric modeling, meshing, spatial curve and shape analysis, comparison, and retrieval. For more information, please visit http://www.cs.sunysb.edu/~xinli. He is a student member of the IEEE and the IEEE Computer Society.

**Yunfan Bao** received the BSc degree (2003) in computer science and engineering from Zhejiang University, China. He is a PhD candidate in the Computer Science Department at Stony Brook University. His research interests include shape modeling, physics-based modeling, animation and human computer interaction. For further information, please visit http://www.cs.sunysb.edu/~ybao. He is a student member of the IEEE and the IEEE Computer Society.

**Xianfeng Gu** received the PhD in computer science from Harvard University in 2003. He is an assistant professor of computer science at Stony Brook University. He won the US National Science Foundation CAREER award in 2004. His research interests are computer graphics, computer vision, and medical imaging. His major works include geometry images, global conformal surface parameterization, manifold splines, and computational conformal geometry. For more information, see http://www.cs.sunysb.edu/~gu. He is a member of the IEEE and the IEEE Computer Society.

**Hong Qin** is associate professor of computer science at the State University of New York at Stony Brook. In 1997, Professor Qin was awarded the US National Science Foundation CAREER Award. In December 2000, he received a Honda Initiation Grant Award. In April 2001, he was selected as an Alfred P. Sloan Research Fellow by the Sloan Foundation. His areas of expertise include geometric modeling, graphics, physics-based simulation, computer aided geometric design, and human-computer interaction. At present, he is an associate editor of *IEEE Transactions on Visualization and Computer Graphics* (TVCG) and he is also on the editorial board of *The Visual Computer (International Journal of Computer Graphics)*. In 2005, he cochaired the 23rd Computer Graphics International Conference (CGI 2005). For further information, please visit http://www.cs.sunysb.edu/~qin. He is a member of the IEEE and the IEEE Computer Society.