

# Harmonic Parameterization by Electrostatics

HE WANG

University of Edinburgh

KIRILL A. SIDOROV

Cardiff University

and

PETER SANDILANDS and TAKU KOMURA

University of Edinburgh

In this article, we introduce a method to apply ideas from electrostatics to parameterize the open space around an object. By simulating the object as a virtually charged conductor, we can define an object-centric coordinate system which we call Electric Coordinates. It parameterizes the outer space of a reference object in a way analogous to polar coordinates. We also introduce a measure that quantifies the extent to which an object is wrapped by a surface. This measure can be computed as the electric flux through the wrapping surface due to the electric field around the charged conductor. The electrostatic parameters, which comprise the Electric Coordinates and flux, have several applications in computer graphics, including: texturing, morphing, meshing, path planning relative to a target object, mesh parameterization, designing deformable objects, and computing coverage. Our method works for objects of arbitrary geometry and topology, and thus is applicable in a wide variety of scenarios.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*

General Terms: Algorithms, Design, Experimentation, Theory

Additional Key Words and Phrases: Coordinates, parameterization, cloth control

---

H. Wang and K. A. Sidorov are joint first authors. Principal theoretical contribution is by K. A. Sidorov and Principal experimental contribution is by H. Wang. This work was supported by EU FP7 TOMSY and EPSRC Standard Grant (EP/H012338/1).

Authors' addresses: H. Wang, School of Informatics, The University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK; K. A. Sidorov, Department of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Roath, Cardiff CF24 3AA, UK; P. Sandilands, T. Komura (corresponding author), School of Informatics, The University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK; email: tkomura@inf.ed.ac.uk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2013 ACM 0730-0301/2013/09-ART155 \$15.00

DOI: <http://dx.doi.org/10.1145/2503177>

## ACM Reference Format:

Wang, H., Sidorov, K. A., Sandilands, P., and Komura, T. 2013. Harmonic Parameterization by electrostatics. *ACM Trans. Graph.* 32, 5. Article 155 (September 2013), 12 pages.

DOI: <http://dx.doi.org/10.1145/2503177>

## 1. INTRODUCTION

Designing and controlling in a nontrivially constrained open space around objects that include concavities is a difficult problem due to the difficulty of efficient parameterization. One of the main problems occurring with previous methods is that the state of the open space is simply described by raw world 3D Cartesian coordinates, rather than in some way relative to the surfaces of the other objects involved in the scene. This makes avoiding collisions and inter-penetrations very difficult. An object-centric state space, based on the spatial relationship between the geometry of different constituent objects, would greatly help to avoid such problems.

One possible solution is to represent configurations in the open space by parameters with respect to the shape of a reference object. The generalized barycentric coordinates (or cage-based coordinates) is a related concept. Using generalized barycentric coordinates, one can first design an intricate scene in a canonical configuration and then edit the scene by manipulating the vertices of the lattice that surrounds the scene. However, in general, generalized barycentric coordinates are designed to parameterize the interior of a closed volume but not for the exterior of the lattice, and so can be unintuitive for this type of control.

In this article, we propose to use ideas from electrostatics to parameterize the state of an object in the open space with respect to the underlying reference object. First, we define an object-centric curvilinear coordinate system which we call Electric Coordinates. We simulate the reference objects in the scene as virtually charged conductors. The field lines and equipotential surfaces of the resulting electric field parameterize the open space that surrounds the objects. The electric potential around a charged object is a harmonic function, which cannot have local extrema except at the boundary. Therefore, the electric field lines (curves of steepest descent of the electric potential) all diverge to points at infinity without being attracted to local extrema. This produces a mapping between the surface points of objects with arbitrary topology and a sphere at infinity.

Also, by using shape-invariant properties of Gauss's law in integral form [Katsikadelis 2002], we can compute a parameter called flux, that quantifies the degree to which the reference object is surrounded by another object (hereafter referred to as deformable object) such as a cloth. The flux can be used as a reference to

direct movements of deformable objects in Electric Coordinates to cover or uncover the surface of a charged object. Control of wrapping maneuvers is a challenging problem in computer animation and robotics [Igarashi et al. 2009; Wang and Komura 2012] when the deformable object has very many degrees of freedom (such as a cloth composed of many particles). By Stokes' theorem [Spivak 1965], simply moving the boundary of the deformable surface in the direction of the fastest increase of the flux through the surface will result in the most efficient movements to cover the reference object.

We believe the electrostatic parameters may have several applications in computer graphics, including path planning, designing deformable objects, spherical mesh parameterization, and guiding complex maneuvers such as wrapping objects by cloth or a bag.

As our method works for objects of arbitrary geometry (convex or nonconvex) and topology (can handle multiple objects as well as objects with holes or handles) it can be robustly applied in various scenarios.

#### Contributions.

- We provide a novel parameterization of the space, based on the concepts of electrostatics.
- We give an abstract, intuitive, shape-insensitive measure of coverage of one object by another.
- We supply a method to synthesize complex wrapping movements without global path planning.

The rest of the article is organized as follows. After reviewing the related work in Section 2, we describe the charge simulation technique and then discuss the electrostatic parameters comprising Electric Coordinates and flux in Section 3. We discuss the advantages and limitations of our method with respect to other methods as well as quantitatively evaluating our approach in Section 4, and after we make concluding remarks in Section 5.

## 2. RELATED WORK

Our work is most closely related to parameterization of the space and its applications to 3D computer graphics, which is reviewed first. As we solve Partial Differential Equations (PDEs) by a Boundary Element Method (BEM), we next review works that make use of PDEs for computer graphics applications and BEM in particular. Finally, we review spatial relationship-based representations that are used in computer animation.

*Volume and Mesh Parameterization.* Generalized barycentric coordinates, such as Harmonic Coordinates [Joshi et al. 2007], Green Coordinates [Lipman et al. 2008], and Mean Value Coordinates [Floater 2003; Ju and Schaefer 2005], represent the location of points in the interior of a closed lattice (cage) as a function of control points' positions. Using generalized barycentric coordinates, animators can easily deform 2D and 3D objects by adjusting the control points of the surrounding cage. Although each approach has its advantages and disadvantages, most are not designed to parameterize the space outside the cage; and those that do [Ju and Schaefer 2005] suffer artifacts (see comparison in Joshi et al. [2007]).

One way to parameterize the outer space of objects is to use distance fields. Distance fields can parameterize the space with respect to a reference object based on the Euclidean distance. This has been applied to, for example, image processing, motion planning, collision detection, and 3D design [Perlin and Hoffert 1989; Schmid et al. 2011]. The mapping between a point and the nearest point on

the surface is discontinuous in a distance field at medial axes, where the distance to different points on the surface is the same (i.e., near medial axes, nearby points in 3D space can map to distant points on the surface). This may be an issue when a smooth continuous parameterization is preferred.

Peng et al. [2004] produce smooth continuous distance fields that parameterize the space in the vicinity (thick shell) of a reference object. They apply their method to define 3D textures and displacement maps within the parameterizable shell. While an improvement over naïve distance fields, the method of Peng et al. [2004] is not capable of continuously parameterizing the entire space: far from the shell their method suffers the same problem as distance fields, because the potential fields in their work can have local extrema to which the solutions of their ODEs (field lines) attract. This is particularly prominent when reference objects have nonconvex geometry. Although our method has some similarities with Peng et al. [2004], there are important fundamental differences (notably, the harmonic nature of our potential fields), due to which our approach does not suffer the aforesaid problems. We further compare our method and that by Peng et al. [2004] in Section 4.

Mesh parameterization is also related to our study as our method produces a mapping between points on the surface of an object and those on a sphere at infinity. As mesh parameterization is a vast research area, we only refer to the core spherical parameterization papers that are strongly related to our work. The readers are referred to Floater and Hormann [2005] and Sheffer et al. [2007] for a more complete review. Spherical parameterization has been intensely researched, and methods based on projected Gauss-Seidel iterations [Gu et al. 2004], barycentric embedding [Gotsman et al. 2003], coarse-to-fine embedding [Shapiro and Tal 1998; Praun and Hoppe 2003], and exterior derivatives [Gu and Yau 2003] have been proposed. Such methods are quantitatively evaluated by criteria based on computation time and distortions of angles, lengths, and areas. An important qualitative criterion is whether they flip triangles or not (produce a one-to-one mapping) [Shapiro and Tal 1998; Praun and Hoppe 2003; Gotsman et al. 2003]. We evaluate our method from these viewpoints in Section 4.

*PDEs and Boundary Element Method.* In computer graphics, PDEs, and especially the Laplace equation, have been widely used for applications such as quadrilateral remeshing [Dong et al. 2005], mesh editing [Au et al. 2007], image editing [Pérez et al. 2003], and generalized barycentric coordinates [Joshi et al. 2007]. In these works, the solutions are computed by Finite Element Methods (FEM) in which the domain is tessellated (divided into small elements) uniformly or hierarchically (e.g., with octrees [Kazhdan et al. 2007]) and then the PDE is solved on the tessellated domain. Such an approach is not suitable for our purposes (solving Laplace equation in the open space) as it would require tessellating the entire infinite space and solving a large-scale linear system whose dimension is the number of elements. As we need to be able to compute the field everywhere in the 3D open space (or a large working subvolume), the size of the grid can be very large in some cases, depending on the working volume and the geometry of the reference object.

On the contrary, we use a Boundary Element Method (BEM), in which the PDEs are rewritten in an integral form. (For an overview of BEMs, see Katsikadelis [2002].) The BEM represents a 3D field by a function basis that has its degrees of freedom attached to a surface. The advantage of the BEM is that we do not need to apply a tessellation as in FEM everywhere in space, because the field can be computed analytically from the boundary parameters. All computations in our method are performed on the tessellated

*boundary* of the space. BEMs are, in general, advantageous when the size of the volume in question is much larger (in terms of the number of elements required to achieve a given accuracy) than the size of the boundary, as is the case in this article. The disadvantage is that the computation of the boundary parameters is costly as it is necessary to solve a dense linear problem. We later show that we can still solve large enough problems for practical applications in Section 4.

*Control Based on Spatial Relationship.* Modeling of spatial relationships has been applied to deformation transfer [Zhou et al. 2010] and motion retargeting [Ho et al. 2010]. Zhou et al. [2010] apply the minimal spanning tree to preserve the relationship between a dress and a character while dancing. Ho et al. [2010] use the Delaunay tetrahedralization to encode the spatial relationship between interacting characters. The structures used in these works are not suitable for parameterizing the continuous configuration space due to their discrete nature. A continuous parameterization is more suitable for synthesizing novel motions.

Ho and Komura [2009] use the Gauss Linking Integral to guide movements of a garment when dressing. A character successfully passes its arms through the sleeves of a shirt in their demo. This approach is successfully applied for controlling robots to put a shirt on humans [Tamei et al. 2011]. In these approaches, the body is represented by articulated 1D links and the surface-surface relationship is not considered. However, modeling surface-surface relationships would be advantageous for animating close interactions between a garment and a character. We are going to seek an approach to quantify such relationships in this article.

In summary, a continuous parameter space based on spatial relationship is most suitable for guiding wrapping-like movements. However, there has been little work that quantifies the surface-surface relationship of a deformable object wrapping around a reference object. This work achieves this.

### 3. ELECTROSTATIC PARAMETERIZATION

The idea behind our method is to simulate the reference object as a charged conductor using a method called charge simulation (see Section 3.1), and use its physical properties to parameterize the open space around the object, and also to compute an abstract and intuitive parameter which represents the spatial relationship between the reference object and a deformable object that surrounds it.

More specifically, electrostatics provides us with the following two important concepts: (1) *An object-centric curvilinear coordinate system:* A curvilinear coordinate system, which we call Electric Coordinates, can be defined using the electric potential and field lines induced by the charged reference object (see Section 3.2). (2) *Coverage:* A physical quantity called flux, which can be computed by Gauss's law (see Section 3.3), represents the coverage of the reference object by the deformable object. This can be used to guide wrapping movements.

The algorithmic overview of the calculation and the pointers to each corresponding section is presented in Figure 1.

#### 3.1 Charge Simulation

In this section, we explain charge simulation, which is a BEM-based approach to compute charge distribution on the conductor that satisfies the required boundary conditions. Charge simulation allows us to solve the Laplace equation by performing computations only on the boundary of the reference object, which then allows us to compute the electric field and potential *everywhere* in the space analytically using the superposition principle.

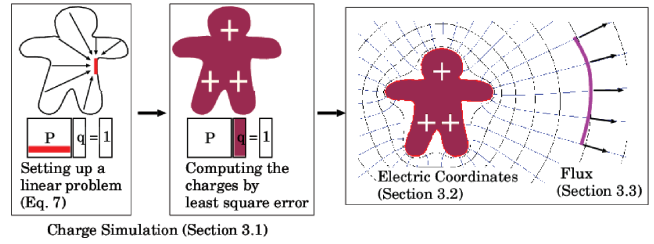


Fig. 1. The algorithmic overview of our approach.

Charge simulation is, in general, a continuous problem. We discretize it by assuming that the objects are formed by a mesh of triangles, each of which has a constant distribution of charge. Shortly, we first describe how this discretization applies to the superposition principle, and then to the charge simulation.

*Superposition Principle.* Let us first describe the computation of the electric field and potential by the superposition principle. The total electric field at some point  $\mathbf{x}$  in space is a vectorial sum of electric fields produced by all point charges. A general continuous formulation, for an arbitrary distribution of charge, can be expressed as a volume integral of charge density over the region of the charge distribution

$$\mathbf{E}(\mathbf{x}) = \int_V d\mathbf{E} = \frac{1}{4\pi\epsilon_0} \int_V \frac{\rho}{r(\mathbf{x})^2} \hat{\mathbf{r}} dV, \quad (1)$$

where  $\rho$  is the charge density being integrated (the amount of charge per unit volume),  $dV$  is the differential volume element,  $\hat{\mathbf{r}}$  is the unit direction vector from  $dV$  to  $\mathbf{x}$ , and  $r$  is the distance between  $\mathbf{x}$  and  $dV$ . The electric potential is defined similarly.

$$P(\mathbf{x}) = \int_V dP = \frac{1}{4\pi\epsilon_0} \int_V \frac{\rho}{r(\mathbf{x})} dV \quad (2)$$

(Note also that the electric field is simply the negative gradient of the potential.)

In our method, we use a discrete representation of charges on the tessellated boundary of an object. Assume that the reference object, simulated as a charged conductor, is represented by a triangulated mesh (on which all of the charge resides). Let  $T_i$  denote the triangles of the mesh. According to the superposition principle, the electric field vector  $\mathbf{E}_{\text{tot}}$ , at an arbitrary point  $\mathbf{x}$  in space, generated by the whole object is the sum of the electric fields generated by individual triangles  $T_i$ .

$$\mathbf{E}_{\text{tot}}(\mathbf{x}) = \sum_i \mathbf{E}(T_i, \mathbf{x}) \quad (3)$$

The preceding also applies to electric potential.

$$P_{\text{tot}}(\mathbf{x}) = \sum_i P(T_i, \mathbf{x}) \quad (4)$$

The electric field and potential generated by an individual triangle are functions of its vertex coordinates and of the distribution of charge on the surface of the triangle. Analytic expressions exist for the electric field and potential due to a uniformly charged triangle [Goto et al. 1992] (also presented in Appendices A.1 and A.2). We found experimentally that this approximation is sufficient in most cases. For a more accurate modeling of charge distribution, Tatematsu et al. [2000, 2002] give expressions for the field due to a 2nd- and 3rd-order charge distribution. For ease of explanation, let us assume constant charge distribution on each triangle. (The principle remains the same for higher-order interpolated charge distribution.)

*Charge Simulation on a Triangle Mesh.* We now describe how the continuous charge simulation problem is converted into a discrete problem. The charge simulation method computes the distribution of charges such that the electric potential is the same everywhere on the surface. We solve this problem in practice by applying two discretizations: (1) assuming the constant charge distribution model on each triangle, (2) we enforce the same potential at each triangle's barycenter by computing the charges carried by all triangles. Shortly we formalize how the continuous nature of the charge simulation problem is converted into a discrete problem.

The charge simulation method computes the distribution of charges as if the object is a thin, charged conductor at equilibrium. On the surface of a conductor, the charge is distributed in such a way that the electric field at each point on the surface is normal to the surface. (Indeed, if the electric field was not normal at some point on the surface, then there would be a nonzero tangential component that would move the charge along the surface, which contradicts the equilibrium assumption.) Or, equivalently, the charge is distributed in such a way that the electric potential at every point on the surface of a conductor (at equilibrium) is the same (recall that the electric field is the negative gradient of the electric potential). The problem of charge simulation is, therefore, to compute charge density  $\rho$  at every probe point  $\mathbf{p}$  (on the surface of the object) such that the resulting field satisfies the preceding properties. Let  $\Omega$  be the surface of the conductor, and let  $\mathbf{x} \in \Omega$  be points of the surface. Then charge simulation amounts to solving the equation

$$\text{const} = P(\mathbf{p} \in \Omega) = \int_{\Omega} \frac{\rho(\mathbf{x})}{|\mathbf{p} - \mathbf{x}|} d\Omega(\mathbf{x}), \text{ for } \rho(\mathbf{x}) \text{ at all } \mathbf{x} \in \Omega. \quad (5)$$

In our approach, we discretize the charge (assume a constant charge distribution on each of the triangles of the mesh), in order that charge simulation can be easily performed by a linear method. Assume that each triangle carries (yet unknown) charge  $q_i$ . Let  $P(T_i, q_i, \mathbf{x})$  be the potential at an arbitrary point  $\mathbf{x}$  in space due to the  $i$ -th triangle  $T_i$  carrying charge  $q_i$ . Further, the potential due to a charged element is proportional to its charge, so  $P(T_i, q_i, \mathbf{x}) = q_i P(T_i, 1, \mathbf{x})$ . Let  $\mathbf{p}_j \in \Omega$  be probe points on the surface  $\Omega$  of the object, each of which set to the barycenter of the  $j$ -th triangle. The potential at these probe points must be the same on the surface of a conductor. Without loss of generality, assume the potential on the surface is 1 "volt". By principle of superposition, for some probe point  $\mathbf{p}_j$  this potential of 1 volt is the sum of potentials due to all triangles of the object.

$$P_{\text{tot}}(\mathbf{p}_j) = \sum_i P(T_i, q_i, \mathbf{p}_j) = \sum_i q_i P(T_i, 1, \mathbf{p}_j) = 1 \quad (6)$$

We need to find all charges  $\{q_i\}$  of individual triangles such that Eq. (6) holds for every probe point  $\mathbf{p}_j$  on the surface of the object. If there are  $n$  triangles, we have  $n$  unknowns  $q_i, i = 1 \dots n$ . To find  $\{q_i\}$  we need to compose a linear system of  $n$  equations, each of the form as Eq. (6) reflecting the fact that the potentials at all probe points  $\mathbf{p}_j$  are equal to 1 volt. The resulting system, with one equation per probe point, has the form

$$\begin{cases} q_1 P(T_1, 1, \mathbf{p}_1) + \dots + q_n P(T_n, 1, \mathbf{p}_1) = 1 \\ \dots \\ q_1 P(T_1, 1, \mathbf{p}_n) + \dots + q_n P(T_n, 1, \mathbf{p}_n) = 1, \end{cases} \quad (7)$$

where  $P(T_i, 1, \mathbf{p}_j)$  is the potential due to the  $i$ -th triangle carrying unit charge (which is computed analytically [Goto et al. 1992; Tatematsu et al. 2000; 2002]; see Appendix A.1 (Eq. (16)) for the expressions) and  $\{q_i\}$  are the unknown charges.

Note that although the analytic expressions for the potential due to an element are exact, the fact that the charge is modeled discretely is only an approximation: in the exact solution to the charge simulation problem, the resulting distribution of charge within triangles will not, in general, be constant. Therefore, there are situations when no exact solution exists for Eq. (7) and the previous system should be, therefore, solved in the least squared sense.

When performing the simulation on low-resolution meshes, it is prudent to increase the density of the mesh to more accurately represent the charge distribution, especially in sharp protruding areas where charge density is higher. We discuss this issue in more detail in Section 4.2.

Once  $\{q_i\}$  are computed, we can calculate the electric field and potential anywhere in the space using the superposition principle (Eq. (3), Eq. (4), Eq. (16), Eq. (20)).

Figure 2 and Figure 3 visualize the distribution of charges over the surfaces of different objects, the resulting potential (in the slices), and the resulting electric fields by plotting field lines  $d\mathbf{x}/dt = \mathbf{E}(\mathbf{x})$ .

### 3.2 Electric Coordinates

Every point in the 3D space can be characterized by its potential and a unique gradient line (field line) on which it lies. We use such physical properties of electrostatics to define our Electric Coordinates. Indeed, the field lines emanating from each point on the surface never intersect<sup>1</sup> and since the potential harmonically decreases with distance from the object, each point along a particular field line has a unique potential (see Figure 4 (right)). Each field line is defined by its starting point on the surface (in some parametric  $(u, v)$  coordinates). Therefore, except at saddle points of the potential (which are discussed shortly), each point in the 3D space can be represented by two orthogonal elements: its electric potential ( $P_x$ ) and the parametric coordinates  $(u, v)$  of the origin of the corresponding field line on the surface of the charged object.

Since electrostatic potential is a harmonic function, it cannot have maxima or minima (except at the boundary). However, isolated saddle points (points of unstable equilibrium, where the electric field is zero and, therefore, no field line may be traced from them) may exist. It can, however, be shown that saddle points cannot form a volume according to the maximum principle [Berenstein and Gay 1997]. Indeed, if there was a region of space where the electric field is everywhere zero, the potential in this region would be constant. According to the maximum principle, the potential must then be constant in the entire domain. Therefore, the potential can either be constant (this is a trivial case, e.g., as inside a conductor) or it may not contain areas of saddle points. Therefore, saddle points are isolated and in practice they can be ignored.

The electric potential  $P_x$  is computed analytically by Eq. (16) in Appendix A.1. The projected point  $S_x$  on the reference object is computed by tracing the field lines from  $\mathbf{x}$  back onto the reference object. Field lines, a common way of depicting the electric fields, are the integral curves of  $d\mathbf{x}/dt = \mathbf{E}(\mathbf{x})$ , or, equivalently, curves of steepest descent of the potential, and so tracing a field line from some starting point  $\mathbf{x}$  is a simple initial value problem.

<sup>1</sup>Electric field lines never intersect: the tangent to an electric field line at each point is in the same direction as the electric field vector at that point (because, by definition, the field lines are the integral curves of  $d\mathbf{x}/dt = \mathbf{E}(\mathbf{x})$ ). If two electric field lines intersected, this would have meant that the electric field (a one-valued smooth vector function) at the intersection point has two different directions, which is impossible. More formally, this follows from the Existence and Uniqueness theorem (Cauchy-Lipschitz).

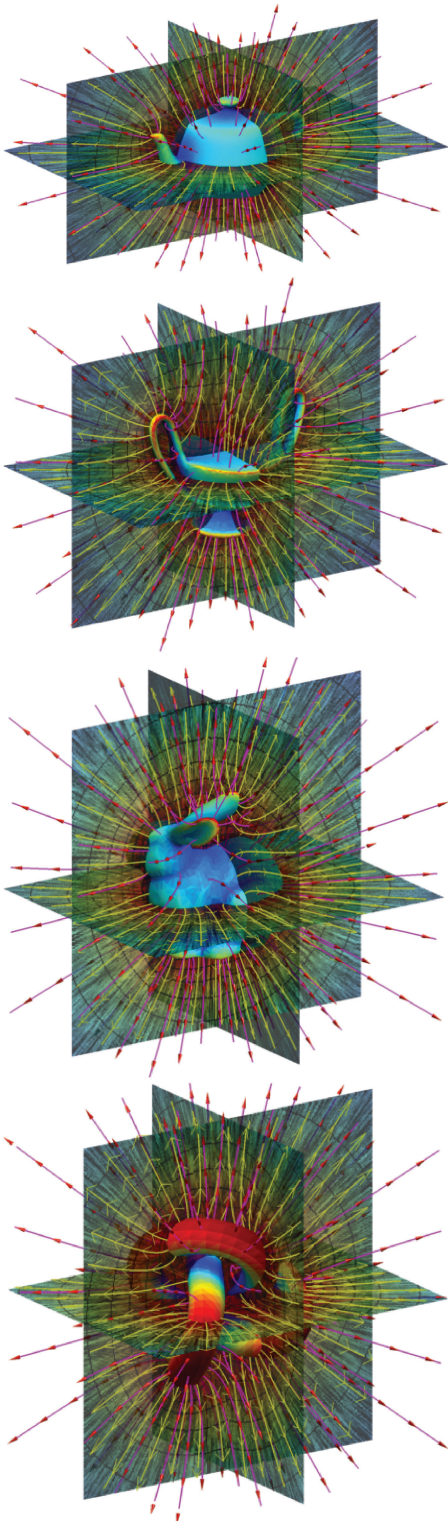


Fig. 2. Electric fields around surfaces. The color of the object represents the charge density (red = high, blue = low). The bunny model is from Stanford 3D Scanning Repository [Stanford Computer Graphics Laboratory 2013]. The vase model is from Princeton Shape Benchmark [Shilane et al. 2004].

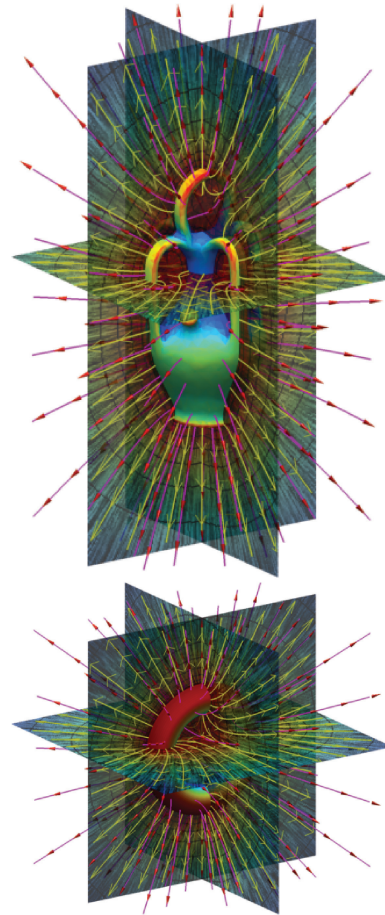


Fig. 3. Electric fields around surfaces (continued). The vase model is from Princeton Shape Benchmark [Shilane et al. 2004].

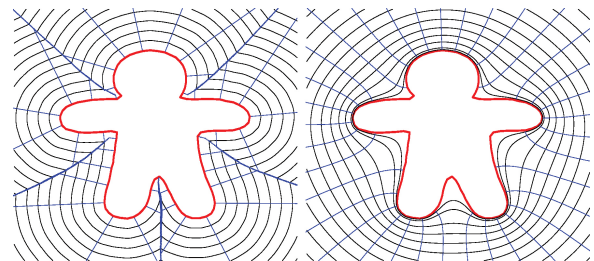


Fig. 4. *Left*: the isosurfaces and the field lines of the distance field. *Right*: the grid (field lines and equipotentials) of our curvilinear Electric Coordinates system.

An analogous parameterization in the case of, for example, distance fields is discontinuous/nonsmooth, in which discontinuities in coverage and nonsmoothness in potential (distance) exist at medial axes for nonconvex objects, as shown in Figure 4 (*left*).

Note also that the electric field inside a charged conductor at equilibrium, which our method simulates, is zero (and the potential is constant). Indeed, if there is a nonzero field inside then the free charge in the conductor would move, which contradicts the equilibrium condition. Because of the zero field, the proposed coordinate



Fig. 5. An example of growing grass along the electric field.

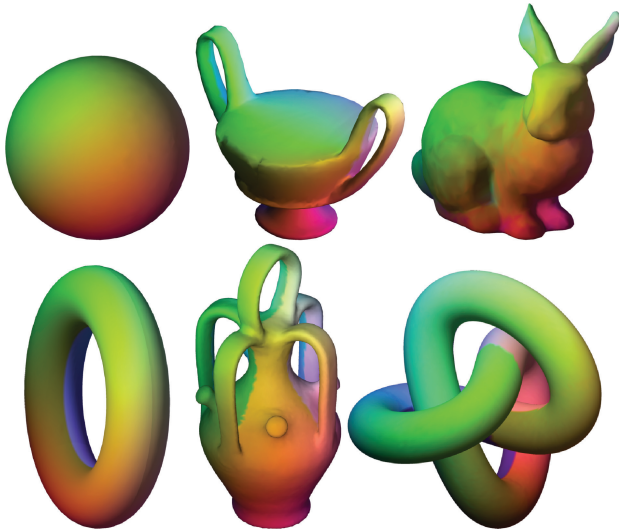


Fig. 6. Mapping surfaces to a sphere. Color indicates the resulting spherical coordinates.

system, based of field lines and equipotential surfaces, is undefined inside an object.

The spatial parameterization by Electric Coordinates is useful for various purposes in computer graphics. The mapping between the 3D positions and the Electric Coordinates is smooth and continuous (except around integral lines passing through saddle points). Such a feature is useful for designing deformable objects in concave constrained space between the charged objects. A simple example of growing grass in constrained concave areas towards an open space without self-collisions is shown in Figure 5: starting from seed points distributed over the ground, we compute the electric field and its tangent plane, and then determine the direction in which to grow.

It is also useful for path planning in such concave areas as collisions with the charged objects can be avoided by simply examining the electric potential. Finally, it is applicable to transferring the configuration of deformable objects which surround a charged object, because a self-collision-free configuration will be mapped to another self-collision-free configuration when the geometry of the charged object is changed.

By tracing the field lines from the surface of a charged object to infinity (in practice, to sufficiently large distance) we can compute a mapping between the surface and a sphere (recall that equipotential

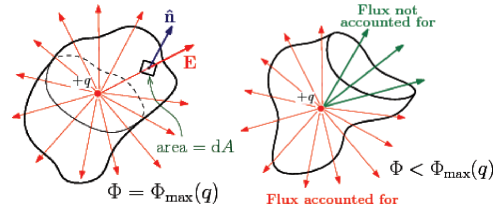


Fig. 7. Illustration of the Gauss's law.

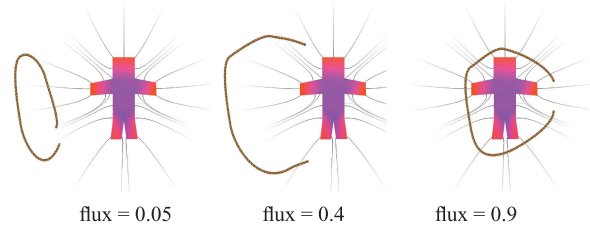


Fig. 8. Different configurations of the deformable object and the reference object, and the corresponding flux values.

surfaces at infinity are spheres). This is illustrated in Figure 6, where color indicates the resulting coordinates on a sphere. For objects with handles, a continuous seamless mapping to a sphere does not exist (because of the saddle points in the middle of the handles). However, our method admirably copes with this scenario also: the seams occur naturally (one per handle) where the field lines diverge around the handles, in aesthetically logical positions (Figure 6 shows several such examples).

### 3.3 Computing Coverage by Gauss's Law

We now explain how to quantify how much the reference object is surrounded by the deformable object. Gauss's law, which states that the total amount of electric flux through any closed surface is proportional *only* to the enclosed electric charge, provides a good way to quantify such coverage. Gauss's law in integral form [Matthews 1998] can be formulated as

$$\Phi = \oint_S \mathbf{E} \cdot d\mathbf{A} = \oint_S \mathbf{E} \cdot \hat{\mathbf{n}} dA = \frac{Q}{\epsilon_0} = \text{const}, \quad (8)$$

where  $\mathbf{E}$  is the electric field being integrated over the surface  $S$  which is surrounding the charged object (with charge  $Q$ ),  $d\mathbf{A}$  is an infinitesimal region of  $S$  (a vector with area  $dA$  pointing in the normal direction  $\hat{\mathbf{n}}$ ), and  $\epsilon_0$  is the electric constant (see Figure 7, left). This means the total electric flux  $\Phi$  through a closed surface  $S$  depends only on the charge  $Q$  enclosed by that surface, and does *not* depend on the *shape* of the surface. If the surface is not closed, the integral in Eq. (8) will not account for all of the flux and the value of  $\Phi$  will be smaller. Since the integral in Eq. (8) can also be computed for open surfaces, it becomes a good measure of how much a surface wraps around a charge. This is illustrated in Figure 7 showing the flux through a closed and an open surface. Note also that the dot product with the surface normal in Eq. (8) accounts for the correct net flux even when a field line penetrates the surface multiple times.

Examples of different configurations in which a deformable object is surrounding the reference object are shown together with the flux value in Figure 8. It can be observed that the more the object is surrounded, the larger the flux. For polygonal meshes, the integral in Eq. (8) can be computed by summing the flux through

all the triangles in the mesh. Analytical expressions for the flux through a triangle do exist [Van Oosterom and Strackee 1983] and are presented in Appendix A.3 for completeness.

*Animating Wrapping Movements Using Flux.* Flux is a continuous parameter which is suitable for guiding movements of deformable objects to wrap and unwrap around the charged object. Here we show examples of such a control by simply moving particles composing the deformable object in the gradient direction of the flux. Note that this is a very high-dimensional path-planning problem (the deformable object typically consists of very many particles) which is difficult to solve by any previous method: any local path-planning algorithm can easily fall into local minima; while for reasons of exponential explosion it is also difficult to apply global path planning, such as random exploration methods.

Let us briefly describe the technical details of the approach. In accordance with Stokes' theorem [Spivak 1965], the flux depends only on the boundary of the surface, and not on the rest of the surface. This allows us to drastically reduce the dimensionality of the control as we only need to actively control the points on the boundary of the surface. Let us define the position of the boundary points by  $\mathbf{c}$ . Given the reference flux increment at each frame, which is represented by  $\Delta\Phi_d$ , the corresponding updates in the boundary points are computed by solving the following least square problem.

$$\arg \min_{\Delta\mathbf{c}} \|\Delta\mathbf{c}\|^2 + \alpha \|\Delta\Phi_d - \mathbf{J}_\Phi \Delta\mathbf{c}\|^2, \quad (9)$$

where  $\mathbf{J}_\Phi$  is the Jacobian matrix of the flux with respect to  $\mathbf{c}$  (experimental results show that flux can be well linearized with respect to  $\mathbf{c}$  for small updates),  $\alpha$  is a proportionality constant for soft constraints (large values of  $\alpha$  mean stricter constraints;  $\alpha$  depends on the scale, tuned experimentally and is set to 100 in our experiments),  $\Delta\Phi_d$  is set to  $\exp(0.1 \times \Phi)$ , where  $\Phi$  is the current flux, and the first term serves to minimize the updates  $\mathbf{c}$  for increasing the stability. The update  $\Delta\mathbf{c}$  can be computed by solving the following linear problem.

$$(\mathbf{I} + \alpha \mathbf{J}_\Phi^T \mathbf{J}_\Phi) \Delta\mathbf{c} = \alpha \mathbf{J}_\Phi^T \Delta\Phi_d \quad (10)$$

After computing  $\Delta\mathbf{c}$ , the  $\mathbf{c}$  is updated by  $\mathbf{c} := \mathbf{c} + \Delta\mathbf{c}$ . Then, a physical simulation is run using a generic cloth simulator to control the rest of the cloth (for example, to enforce nonstretchability) for a single step while constraining  $\mathbf{c}$ , and the state of the whole system is updated. This procedure is repeated until the bag fully wraps around the object.

Snapshots of applying this control are shown in Figure 9. Bag models are automatically controlled to wrap around different polygonal models including the Stanford Bunny and the Armadillo Man. Even in a case where the bag starts from a folded state, it automatically unfolds, directs its opening towards the reference object, and wraps around it while adapting the shape of the opening to the geometry of the object.

This simple control scheme can be applied for robot control and computer animation. Although the abstract and shape-insensitive nature of coverage increases its applicability, sometimes it can be a disadvantage as the details cannot be controlled. In such a case, additional positional constraints can be given by the user for finer control.

## 4. DISCUSSION AND EVALUATION

In this section, we first compare our method with previous related methods (Section 4.1). Next, we evaluate the precision of our charge simulation approach (Section 4.2). Finally, we discuss the computational costs and complexity of our method (Section 4.3).



Fig. 9. Applications of flux-driven control for wrapping. The Armadillo Man model is from Stanford 3D Scanning Repository [Stanford Computer Graphics Laboratory 2013].

### 4.1 Comparison with Previous Methods

We compare the electric field approach with other methods for coverage computation, parameterization of the outer space, and spherical mesh parameterization.

*Parameterization Using Distance Fields.* One alternative to compute the coverage and parameterize the outer space is to use the distance field. Igarashi et al. [2009] evaluate the coverage by testing if the normal vectors intersect the deformable objects, which is similar to using the distance field. Schmid et al. [2011] propose a modeling tool for designing on the outer side of closed surfaces. The distance field is not suitable for our purposes as it cannot handle nonconvex and/or multiple objects well. Singular points, lines, or surfaces are going to exist, which results in the measure of coverage being discontinuous when the surface in question crosses the singular area (see Figure 4, left). As the distance field is not harmonic, it cannot be used to benefit from the local path planning to guide the deformable object.

*Parameterization Using Other Potential-Based Methods.* An alternative for parameterizing the space around the object is the method by Peng et al. [2004]. They consider potential fields around an object of the form

$$P(\mathbf{x}) = \int_{\Omega} |\mathbf{x} - \mathbf{y}|^{-p} d\mathbf{y}, \quad (11)$$

where  $\mathbf{x}$  is an arbitrary point in space at which the potential is computed,  $\mathbf{y}$  is a point on the surface, and  $p$  is the rate of fall off. They take advantage of the fact that for  $p > 1$  in 3D and for  $p > 0$  in 2D the resulting field (gradient of the potential) is normal to the surface at the surface, and, like in this article, trace the field lines to find a shell of small width  $h$  around the surface.

A fundamental difference between Peng et al. [2004] and our method is that we can establish field-based coordinates, while this is not always possible in Peng et al. [2004]. With our method, the entire space can be parameterized. This is because the electric potential is a harmonic function, which cannot have local extrema, and monotonously diminishes from the surface to infinity (field lines start at the surface and go to infinity). The fields in Peng et al. [2004] can have local extrema (see Figure 10, left) and therefore

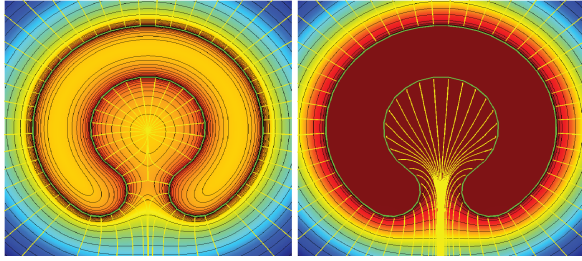


Fig. 10. *Left*: The field computed by Peng et al. (note the local maximum of the potential to which field lines attract). *Right*: The field computed by our method (field lines always escape to infinity).

Table I. Comparison of Spherical Mesh Parameterization Using Our Method and That of Gu et al. [2004]

Mesh	Method	Time (s)	Stretch	Edge	Angle	Area
Bunny*	our	543	16.12	0.3530	0.3700	0.4780
	Gu	4.54	13.43	0.4080	0.1580	0.6680
Suzanne	our	488	12.52	0.2892	0.4188	0.3457
	Gu	1.05	12.23	0.3360	0.0657	0.4820
Dolphin	our	1491	22.97	0.4636	0.5763	0.3737
	Gu	8.89	12.33	0.6586	0.2759	0.9319

\*Bunny model contains 1839 vertices, 3674 triangles; Suzanne model: 1687 vertices, 3370 triangles; Dolphin model: 3561 vertices, 7118 triangles.

cannot be used to define coordinates. Compare with our method in the same situation (Figure 10, *right*).

Also, Gauss's law only holds for very special fields. For Gauss's law to hold, the field strength must fall off exactly as  $1/r^{d-1}$ , where  $r$  is distance and  $d$  is the dimensionality of the space. (Informally: in  $\mathbb{R}^d$ , the surface area grows as power  $d-1$  of linear scale. Therefore, for the Gauss's law to hold, scaling a surface surrounding a charge by a factor  $s$  scales the area by  $s^{d-1}$  and the field strength at the surface must therefore scale by  $1/s^{d-1}$ , as is the case with electric fields.) However, the field lines of Peng et al. [2004] only become perpendicular to the surface (or, equivalently, the surface of the object becomes an equipotential) when the potential fall-off rate  $p$  in Eq. (11) is strictly greater than  $d-2$  and the fall-off rate of the corresponding field is strictly greater than  $d-1$ . As the result, with their method it is not possible to simultaneously define a measure of coverage and the curvilinear coordinate system as we can do in this article.

*Spherical Surface Parameterization.* As our approach also provides a mapping between the object surface and a sphere at infinity, we can compare our approach with existing methods for spherical parameterization. Here we show examples of applying our method to standard mesh models and compare it with the conformal mapping approach [Gu et al. 2004] for reference. Results of mapping bunny, Suzanne, and dolphin models are visualized in Figure 11. It can be observed that area distortion effects are less obvious in electric field-based parameterization although the polygons are sometimes more skewed. It can be observed that our approach maps the surface area that is exposed to the outer space to larger area and the concave area to smaller space. Such effects can be observed at the neck of the bunny and origin of the ears of the bunny and Suzanne.

Results of comparing the the total  $L^2$  stretch error, and absolute distortion of triangle edge lengths, areas, and angles per unit (see Yoshizawa et al. [2004] for the definition of these measures) are presented in Table I. The angle distortion by the electric field-based parameterization is larger than that by the conformal

mapping, although the area and edge length distortion is smaller. Unfortunately, the computation time is significantly larger in our method as it requires solving a large-scale dense linear problem, which takes up much of the time.

In terms of qualitative comparison, our method produces a one-to-one mapping between the object and a sphere when there are no saddle points. Indeed, our system does not flip any triangles in the presented examples although the conformal mapping approach fails to keep the validity in some situations. When there are saddle points, such as at the middle of handles, seams will be produced at lines on the surface that converge to saddle points (the seams in Figure 6). By cutting the surface at such seams, it will be possible to produce a spherical mapping that does not result in triangle flipping. Indeed, another advantage of our method is that the seams due to handles typically occur at aesthetically natural locations.

One main advantage of our method compared to existing spherical parameterization methods is that it produces collision-free trajectories between the corresponding points on the surface and the sphere at infinity. Such trajectories can be used for morphing objects into other objects while avoiding self-collisions. This can be accomplished by morphing both objects to the sphere (or some other potential isosurface), thus establishing correspondence. Another possible application is texture mapping: textures on the sphere can be mapped onto the object as well as other objects in the open space, such as the cloth worn on a character.

## 4.2 Precision of Charge Simulation

In this research, we approximate the continuous distribution of charges discretely, by assuming constant charge distribution per triangle, in order to simplify computations. Although analytical solutions are known for electric field and potential due to higher-order charge distribution on triangles [Tatematsu et al. 2000, 2002], these analytical forms are pages-long and impractical from the implementation point of view. We found that for all practical purposes, our assumption (using analytical expressions from Goto et al. [1992] also given in Appendix A for completeness) provides a good approximation.

Using an Armadillo Man of medium (Figure 12(a)) and low polygon count (Figure 12(c)), the isosurfaces of the electric potential at 1 volt are computed and visualized in Figure 12(b) and (d). It can be observed that the isosurface by the medium-resolution model well approximates the original shape while the low-count version fails to reproduce the sharp parts such as the ears and the tail. This is the artifact of discrete charge distribution. In such a case, the mesh may need to be subdivided a few levels until the isosurface well approximates the original surface. This is illustrated in Figure 12(e), showing the isosurface produced by a mesh subdivided twice.

One way to quantitatively evaluate the distribution of charges is to compute the mean square error of the potential at the surface of the object [Malik 1989]. This is a feasible approach, as the potential on the surface uniquely defines the field everywhere else in space. In Figure 13, we present the 1 volt isosurfaces superimposed with meshes at different resolutions. Further, at each resolution we measure the charge simulation error by measuring the deviation of potential from 1 volt at points densely (uniformly randomly) sampled on the surface of the meshes (Figure 13, below). As expected, the deviation is larger in the coarse model, and smaller at higher resolutions. The deviation from 1 volt is higher at long triangles that are adjacent to extremities (sharp protruding areas). Where high accuracy is required, it may be prudent to subdivide the mesh adaptively, increasing the resolution in areas where the deviation is the largest.



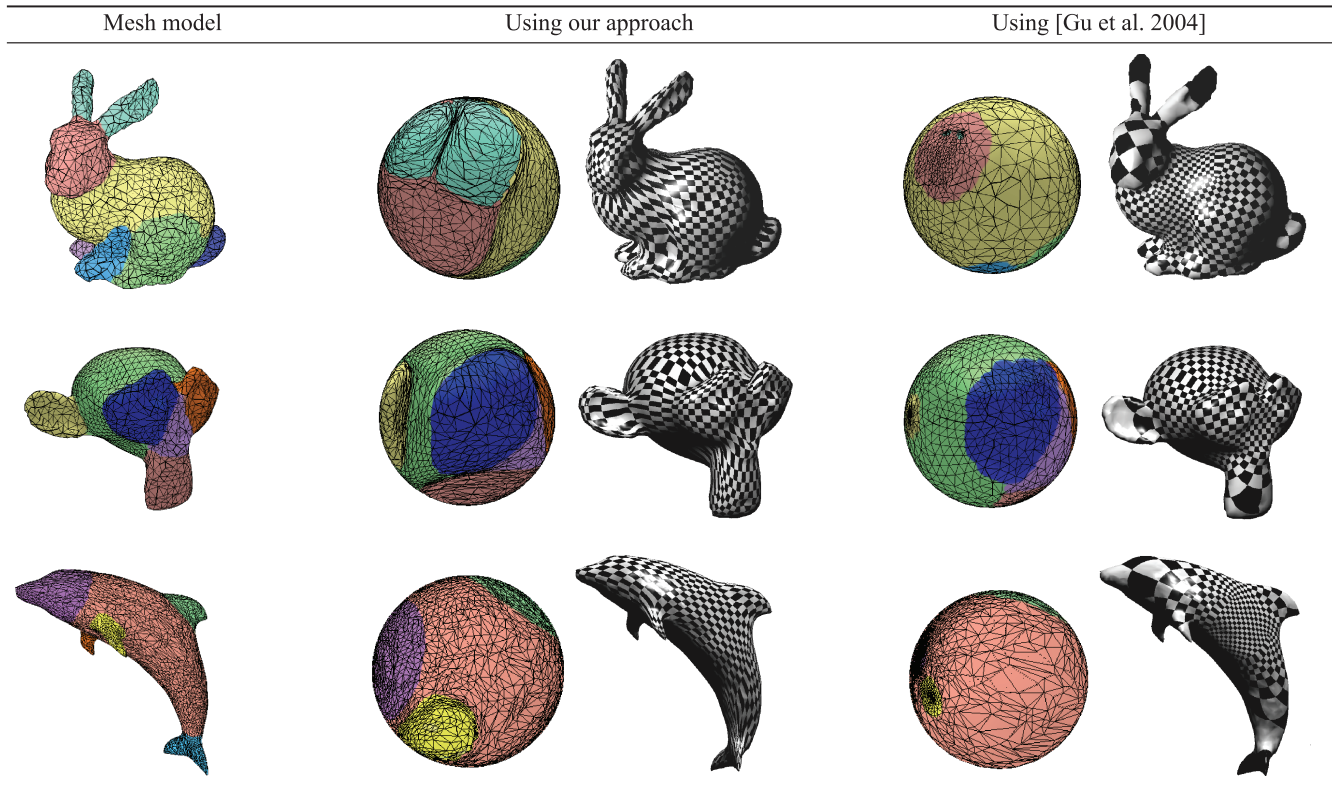


Fig. 11. Visualization of mapping objects onto a sphere, using our approach and that of Gu et al. [2004]. The Suzanne model is from Blender [Blender Foundation 2013] and the dolphin model is from Princeton Shape Benchmark [Shilane et al. 2004].

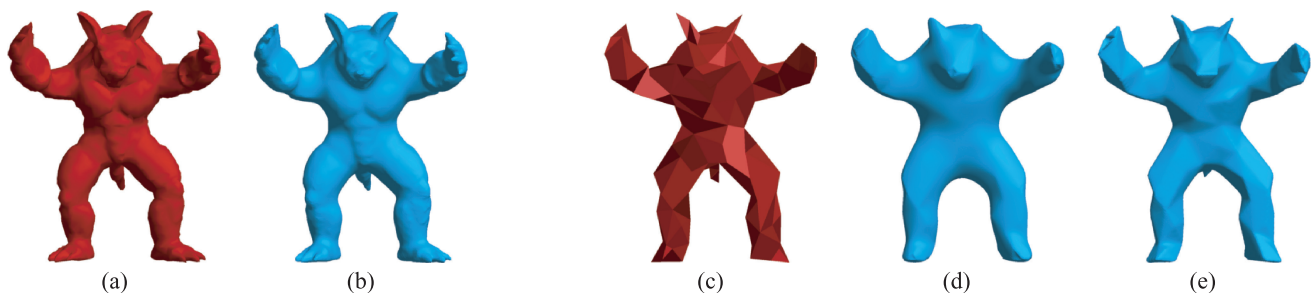


Fig. 12. Isosurfaces of Armadillo: (a) a 7500 triangles model; (b) its isosurface at potential 1 volt; (c) a 300 triangles low-resolution mesh; (d) its isosurface at potential 1 volt; (e) each triangle in mesh (c) has been subdivided into 16 triangles (preserving the original shape) and the isosurface at 1 volt after subdividing is shown.

It is important to note that although our charge simulation is an approximation, the rest of the arguments in this article apply *exactly*. Indeed, that charge distribution is approximate merely means that the equipotential surface of 1 volt does not exactly coincide with the surface of the object. The resulting potential everywhere in space is still exactly harmonic and all the claims of our approach are satisfied exactly.

### 4.3 Complexity and Computational Costs

Let  $n$  be the number of triangles in the charged (reference) object. Complexity of computing electrostatic quantities at an arbitrary point in space (electric field by Eq. (3), potential by Eq. (4)) by

summation over  $n$  triangles of the mesh is  $O(n)$ . Computing the flux by Eq. (8) through another surface of  $m$  triangles is, therefore,  $O(mn)$ . Charge simulation consists of two steps: composing the system of equations (complexity is  $O(n^2)$ ) and solving the dense linear system (complexity is  $O(n^3)$ , but see the following). State updates in wrapping animation (Eq. (10)) with  $k$  control parameters have complexity  $O(k)$ , where each step requires computing the gradient of the flux, making it  $O(kmn)$ .

The charge simulation is costly as the coefficient matrix in Eq. (7) is dense. However, this needs to be done only once for static reference objects. In animation scenarios, when the shape of the reference objects changes gradually between frames, its computation can be significantly accelerated by an iterative solver (e.g., Gauss-Seidel)

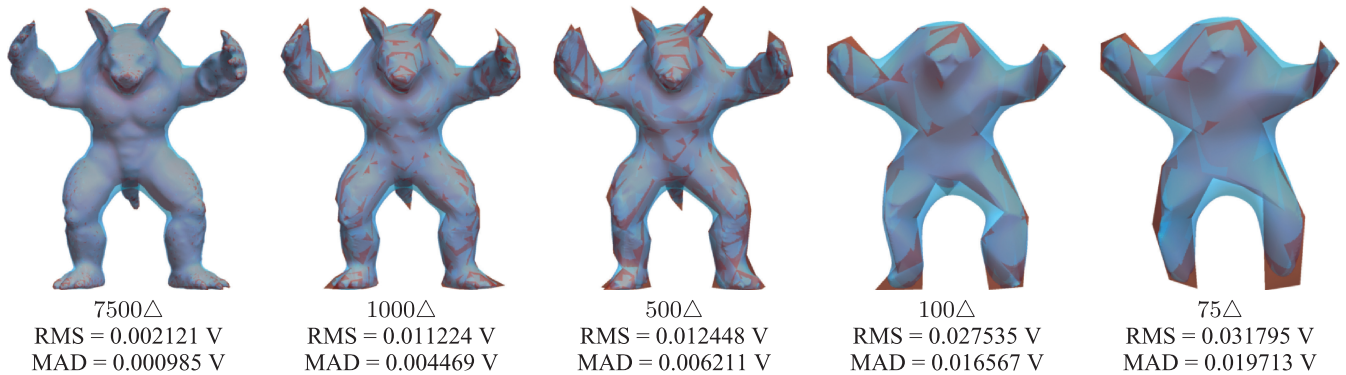


Fig. 13. Errors in charge simulation due to mesh discretization. Armadillo model (red) at various resolutions ( $\Delta$ ) superimposed with the equipotential surface of 1 volt (blue). Below: the root-mean-square and mean-absolute-difference errors from 1 volt at uniformly randomly sampled 300K points at the surface of the object.

Table II. Performance in Guiding a Cloth to Wrap a Reference Object

Scene	<i>def</i>	<i>ref</i>	<i>charge</i>	<i>J</i>	<i>lin</i>
fish	2176	288	0.149	0.076	0.007
bag	868	303	0.165	0.021	0.003

*def, ref*: the triangle count in the deformable and the reference object, *charge, J, lin*: time (in *s*) for charge simulation, calculating the Jacobians, and solving the linear problem, respectively.

using the solution from the previous frame as an initial guess. If the difference between frames is small, only a few ( $w$ ) iterations would suffice making the solution  $O(wn^2)$ .

The computational time required for computing the animation of wrapping the demo using one core of a Core i7 2.67 GHz CPU are shown in Table II. We use UMFPACK [Davis 2004] with GotoBLAS [Goto and Van De Geijn 2008] for solving the sparse linear problem in Eq. (10). We use PhysX 2.84 [NVIDIA 2013] for the physical simulator needed for simulating the wrapping cloth in Section 3.3. Note also that computing electric quantities (summation over triangles) can be trivially delegated to GPU to gain further performance. We have implemented the computation of the field, potential, flux, as well as tracing the field lines using CUDA, and by running the system on a GPU with 384 cores (NVIDIA GeForce GTX 560 Ti), the speedup factor is approximately 20 compared to computing on CPU.

**Limitations.** Only the electric potential is explicitly defined in the Electric Coordinates. Parameterization of the points in the same equipotential surface can be only done by first parameterizing the surface points of the object and then tracing the field lines using integration.

Additionally, a small movement in 3D space can be mapped to large movements on the surface in concave areas (Figure 14, *left*); and, conversely, a small movement on the surface of the object at a sharp convex area can be mapped to large movements in the 3D space (Figure 14, *middle*). Finally, field lines deflect abruptly around saddle points at which the field is exactly zero (e.g., right between the circles in Figure 14, *right*).

## 5. CONCLUSION AND FUTURE WORK

In this article, we have proposed a method for spatial parameterization using electrostatics. First, we can compute a coordinate system

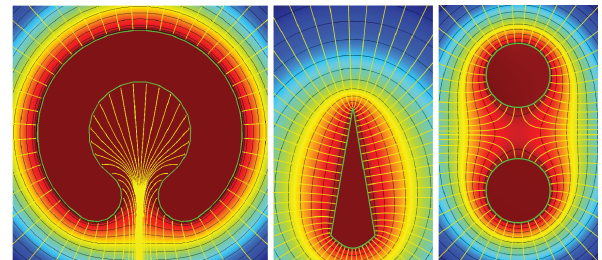


Fig. 14. *Left*: The field around a cavity. *Middle*: The field near a sharp object. *Right*: Field lines around a saddle point.

which we call Electric Coordinates, which can uniquely parameterize arbitrary points in the space. Second, we can compute the flux which can be used to quantify the coverage of a reference object by a surrounding deformable object.

The body-centric Electric Coordinates can be applied for modeling and editing scenes that involve deformable objects such as vegetation as it can parameterize even very constrained spaces. Layered representations [McCann and Pollard 2009; Igarashi and Mitani 2010] which are currently only applicable for 2D instances can be applied to 3D scenes by using the electric potential as a measure of depth. We have also shown that the flux can be applied to synthesizing animations of wrapping maneuvers.

The mapping of the object of arbitrary topology to a sphere at infinity can have several applications, including morphing and texture mapping. It will also be possible to produce a mesh structure which is topologically homeomorphic to a sphere from several mesh structures which are adjacent to one another by tracking the isosurface of a user-specified electric potential which is slightly below 1 volt, in a way analogous to metaballs [Blinn 1982]. This can be useful for CAD applications as it will allow users to synthesize new objects from primitive building blocks by smoothly combining them together.

One interesting application area of our method is robotics. For tasks that involve wrapping and grasping, our framework provides collision-free paths as well as a good measure of how much the reference object has been “wrapped” or “grasped” by a bag or a hand model, respectively. In order to robustly estimate the state of the relationships from noisy real-world data, it may be necessary to use statistical approaches, which is an interesting direction to pursue.

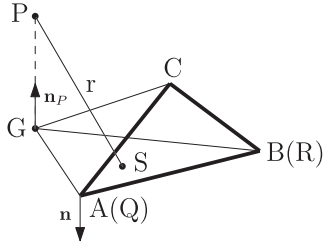


Fig. 15. Computing the potential due to a charged triangle [Goto et al. 1992].

## APPENDIX: COMPUTING ELECTRIC POTENTIAL, FIELD, AND FLUX

Here we describe how to analytically compute the electric potential [Goto et al. 1992] and field around a charged triangular element, as well as the flux through a triangle [Van Oosterom and Strackee 1983] for the case of  $\mathbb{R}^3$ .

### A.1 Potential

Let A, B, C be the vertices of a triangle  $\triangle ABC$  carrying uniformly distributed charge  $q$ . We are interested in computing the electric potential at an arbitrary point P in space.

In the following, let  $\mathbf{x}$  denote the radius vector of a point X, and let  $\overrightarrow{XY} = \mathbf{y} - \mathbf{x}$  denote a vector from point X to point Y.

Let  $\mathbf{n}$  be the unit normal to the triangle.

$$\mathbf{n} = (\overrightarrow{AC} \times \overrightarrow{AB}) / |\overrightarrow{AC} \times \overrightarrow{AB}| \quad (12)$$

Let G be the projection of point P onto the plane of the triangle:  $\mathbf{g} = \mathbf{p} - h\mathbf{n}$ , where  $h = \mathbf{n} \cdot \overrightarrow{AP}$ . Computing the potential in  $\mathbb{R}^3$  involves integrating

$$I = \int_{\triangle ABC} \frac{1}{r} dS, \quad (13)$$

where  $r$  is the distance between P and  $S \in \triangle ABC$ . In Goto et al. [1992] this integral is computed as the sum of integrals over three triangles (see Figure 15)

$$I = I_{\triangle ABG} + I_{\triangle BCG} + I_{\triangle CAG}. \quad (14)$$

Denote the preceding terms generically by  $I(Q, R, G)$ , for QR = AB, BC, or CA. Let  $\sigma = (\overrightarrow{QG} \times \overrightarrow{RG}) \cdot \mathbf{n}$ , which is twice the signed area of  $\triangle QRG$ . Integration yields

$$\begin{aligned} I(Q, R, G) &= \frac{\sigma \log(N/D)}{|\overrightarrow{RQ}|} \\ &+ |h| \arctan \left( \frac{\sigma(\overrightarrow{RQ} \cdot \overrightarrow{RG})(|h| - |\overrightarrow{RP}|)}{\sigma^2 |\overrightarrow{RP}| + |h|(\overrightarrow{RQ} \cdot \overrightarrow{RG})^2} \right) \\ &+ |h| \arctan \left( \frac{\sigma(\overrightarrow{QR} \cdot \overrightarrow{QG})(|h| - |\overrightarrow{QP}|)}{\sigma^2 |\overrightarrow{QP}| + |h|(\overrightarrow{QR} \cdot \overrightarrow{QG})^2} \right), \end{aligned} \quad (15)$$

where  $N = \overrightarrow{RQ} \cdot \overrightarrow{RP} + |\overrightarrow{RQ}| |\overrightarrow{RP}|$ , and  $D = \overrightarrow{RQ} \cdot \overrightarrow{QP} + |\overrightarrow{RQ}| |\overrightarrow{QP}|$ . Additionally, when either  $\sigma = 0$  or  $N = 0$  or  $D = 0$  the expression  $I(Q, R, G)$  should be evaluated to 0.

The potential at point P can now be expressed as

$$P(P) = |I(A, B, G) + I(B, C, G) + I(C, A, G)|. \quad (16)$$

### A.2 Electric Field

The electric field at an arbitrary point is simply the (negative) gradient of the potential. By differentiating Eq. (15) we derive the expressions for the electric field (maintaining the previous notation).

The gradient of the first term in Eq. (15) is

$$\begin{aligned} \Theta &= \frac{\overrightarrow{RQ} \times \mathbf{n}}{|\overrightarrow{RQ}|} \log \left( \frac{N}{D} \right) \\ &+ \sigma \left( \left( \frac{\overrightarrow{RP}}{|\overrightarrow{RP}|} + \frac{\overrightarrow{RQ}}{|\overrightarrow{RQ}|} \right) / N - \left( \frac{\overrightarrow{QP}}{|\overrightarrow{QP}|} + \frac{\overrightarrow{RQ}}{|\overrightarrow{RQ}|} \right) / D \right). \end{aligned} \quad (17)$$

The gradients of the second and the third term in Eq. (15) are similar in form and we express them with parameters  $(t, \mathbf{k})$

$$\begin{aligned} \Psi(t, \mathbf{k}) &= \mathbf{n}_p \arctan \left( \frac{\sigma b t}{v} \right) \\ &+ \frac{\mathbf{n}_p \cdot \mathbf{k}}{\sigma^2 b^2 t^2 + v^2} \left( \sigma t(\sigma^2 + t^2) (\mathbf{n}_p |\mathbf{k}| - \frac{\mathbf{k}}{|\mathbf{k}|} (\mathbf{n}_p \cdot \mathbf{k})) \right. \\ &\left. + b(\overrightarrow{RQ} \times \mathbf{n})t - \sigma \overrightarrow{RQ} (t^2 (\mathbf{n}_p \cdot \mathbf{k}) - \sigma^2 |\mathbf{k}|) \right), \end{aligned} \quad (18)$$

where  $\mathbf{n}_p = \overrightarrow{GP} / |\overrightarrow{GP}|$  is the unit normal to  $\triangle ABC$  in the direction of point P; subexpressions  $v = \sigma^2 |\mathbf{k}| + t^2 (\mathbf{n}_p \cdot \mathbf{k})$ , and  $b = \mathbf{n}_p \cdot \mathbf{k} - |\mathbf{k}|$ . The gradient of the entire Eq. (15) then becomes

$$\mathbf{J}(Q, R, G) = \Theta + \Psi(\overrightarrow{RQ} \cdot \overrightarrow{RG}, \overrightarrow{RP}) - \Psi(\overrightarrow{RQ} \cdot \overrightarrow{QG}, \overrightarrow{QP}), \quad (19)$$

and finally the electric field at an arbitrary point P is

$$\mathbf{E}(P) = -(\mathbf{J}(A, B, G) + \mathbf{J}(B, C, G) + \mathbf{J}(C, A, G)). \quad (20)$$

### A.3 Flux

To compute the electric flux through a triangle  $\triangle ABC$  due to an electric field from a unit point charge at point P, we use the expression from Van Oosterom and Strackee [1983]. Let  $\mathbf{a} = \overrightarrow{AP}$ ,  $\mathbf{b} = \overrightarrow{BP}$ ,  $\mathbf{c} = \overrightarrow{CP}$ . Then the flux  $\Phi(P)$  is found from the expression<sup>2</sup>

$$\tan \frac{\Phi}{2} = \frac{(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}}{|\mathbf{a}| |\mathbf{b}| |\mathbf{c}| + (\mathbf{a} \cdot \mathbf{b}) |\mathbf{c}| + (\mathbf{a} \cdot \mathbf{c}) |\mathbf{b}| + (\mathbf{b} \cdot \mathbf{c}) |\mathbf{a}|}. \quad (21)$$

Note that this is the same as the solid angle around P subtended by  $\triangle ABC$ .

To approximate the flux through a triangle due to another charged triangle, we use Gaussian quadrature to approximate the charged triangle with several point charges.

### ACKNOWLEDGMENTS

The authors thank the reviewers for their constructive comments. They also thank Shin Yoshizawa for the discussion and advice about spherical parameterization. The bunny and armadillo models are from Stanford 3D Scanning Repository [Stanford Computer Graphics Laboratory 2013]. The vase and dolphin models are from Princeton Shape Benchmark [Shilane et al. 2004]. The Suzzane model is exported from the software Blender [Blender Foundation 2013].

<sup>2</sup>Implementation hint: the `atan2` function will take care of the correct sign.

## REFERENCES

- AU, O. K.-C., FU, H., TAI, C.-L., AND COHEN-OR, D. 2007. Handle-aware isolines for scalable shape editing. *ACM Trans. Graph.* 26, 3.
- BERENSTEIN, C. A. AND GAY, R. 1997. *Complex Variables: An Introduction*. Springer.
- BLENDER FOUNDATION. 2013. Blender. <http://www.blender.org/>.
- BLINN, J. F. 1982. A generalization of algebraic surface drawing. *ACM Trans. Graph.* 1, 3, 235–256.
- DAVIS, T. A. 2004. Algorithm 832: UMFPACK, an unsymmetric pattern multifrontal method. *ACM Trans. Math. Softw.* 30, 2, 196–199.
- DONG, S., KIRCHER, S., AND GARLAND, M. 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aid. Geom. Des.* 22, 5, 392–423.
- FLOATER, M. AND HORMANN, K. 2005. Surface parameterization: A tutorial and survey. <http://graphics.stanford.edu/courses/cs468-05-fall/Papers/param-survey.pdf>.
- FLOATER, M. S. 2003. Mean value coordinates. *Comput. Aid. Geom. Des.* 20, 1, 19–27.
- GOTO, E., SHI, Y., AND YOSHIDA, N. 1992. Extrapolated surface charge method for capacity calculation of polygons and polyhedra. *J. Comput. Phys.* 100, 1, 105–115.
- GOTO, K. AND VAN DE GEIJN, R. 2008. High-performance implementation of the level-3 blas. *ACM Trans. Math. Softw.* 35, 1, 1–14.
- GOTSMAN, C., GU, X., AND SHEFFER, A. 2003. Fundamentals of spherical parameterization for 3d meshes. *ACM Trans. Graph.* 22, 3, 358–363.
- GU, X., WANG, Y., CHAN, T. F., THOMPSON, P. M., AND YAU, S.-T. 2004. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* 23, 8, 949–958.
- GU, X. AND YAU, S.-T. 2003. Global conformal surface parameterization. In *Proceedings of the Symposium on Geometry Processing*.
- HO, E. S. L. AND KOMURA, T. 2009. Character motion synthesis by topology coordinates. *Comput. Graph. Forum* 28, 2.
- HO, E. S. L., KOMURA, T., AND TAI, C.-L. 2010. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.* 29, 4.
- IGARASHI, T. AND MITANI, J. 2010. Apparent layer operations for the manipulation of deformable objects. *ACM Trans. Graph.* 29, 4.
- IGARASHI, Y., IGARASHI, T., AND SUZUKI, H. 2009. Interactive cover design considering physical constraints. *Comput. Graph. Forum* 28, 7.
- JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOCKI, T. 2007. Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3.
- JU, T. AND SCHAEFER, S. 2005. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 561–566.
- KATSIKADELIS, J. T. 2002. *Boundary Elements: Theory and Applications*. Elsevier.
- KAZHDAN, M., KLEIN, A., DALAL, K., AND HOPPE, H. 2007. Unconstrained isosurface extraction on arbitrary octrees. In *Proceedings of the 5th Eurographics Symposium on Geometry Processing*. 125–133.
- LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. *ACM Trans. Graph.* 27, 3.
- MALIK, N. H. 1989. A review of the charge simulation method and its applications. *IEEE Trans. Electr. Insul.* 24, 1, 3–20.
- MATTHEWS, P. C. 1998. *Vector Calculus*. Springer.
- MCCANN, J. AND POLLARD, N. S. 2009. Local layering. *ACM Trans. Graph.* 28, 3.
- NVIDIA. 2013. PhysX. [http://www.nvidia.com/object/physx\\_new.html](http://www.nvidia.com/object/physx_new.html).
- PENG, J., KRISTJANSSON, D., AND ZORIN, D. 2004. Interactive modeling of topologically complex geometric detail. *ACM Trans. Graph.* 23, 3, 635–643.
- PEREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph.* 22, 3, 313–318.
- PERLIN, K. AND HOFFERT, E. M. 1989. Hypertexture. In *Proceedings of the 16th Annual ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*. 253–262.
- PRAUN, E. AND HOPPE, H. 2003. Spherical parametrization and remeshing. *ACM Trans. Graph.* 22, 3, 340–349.
- SCHMID, J., SENN, M. S., GROSS, M., AND SUMNER, R. W. 2011. Overcoat: An implicit canvas for 3d painting. *ACM Trans. Graph.* 30, 4, 28:1–28:10.
- SHAPIRO, A. AND TAL, A. 1998. Polygon realization for shape transformation. *The Vis. Comput.* 14, 8–9, 429–444.
- SHEFFER, A., HORMANN, K., LEVY, B., DESBRUN, M., ZHOU, K., PRAUN, E., AND HOPPE, H. 2007. Mesh parameterization: Theory and practice. In *ACM/SIGGRAPH Course Notes*.
- SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The princeton shape benchmark. In *Proceedings of the IEEE International Conference on Shape Modeling Applications*. 167–178.
- SPIVAK, M. 1965. *Calculus on Manifolds*. Vol. 1, Addison-Wesley.
- STANFORD COMPUTER GRAPHICS LABORATORY. 2013. The stanford 3d scanning repository. <http://graphics.stanford.edu/data/3Dscanrep/>.
- TAMEI, T., MATSUBARA, T., RAI, A., AND SHIBATA, T. 2011. Reinforcement learning of clothing assistance with a dual-arm robot. In *Proceedings of the 11th IEEE/RAS International Conference on Humanoid Robots*. 733–738.
- TATEMATSU, A., HAMADA, S., AND TAKUMA, T. 2000. Analytic expressions of potential and electric field generated by a triangular surface charge with second-order charge density. *Trans. IEE Japan* 120-A, 8/9, 853–854.
- TATEMATSU, A., HAMADA, S., AND TAKUMA, T. 2002. Analytical expressions of potential and electric field generated by a triangular surface charge with a high-order charge density distribution. *Electr. Engin. Japan* 139, 3, 9–17.
- VAN OOSTEROM, A. AND STRACKEE, J. 1983. The solid angle of a plane triangle. *IEEE Trans. Biomed. Engin.* 30, 2, 125–126.
- WANG, H. AND KOMURA, T. 2012. Manipulation of flexible objects by geodesic control. *Comput. Graph. Forum* 31, 2.
- YOSHIZAWA, S., BELYAEV, A., AND SEIDEL, H.-P. 2004. A fast and simple stretch-minimizing mesh parameterization. In *Proceedings of the International Conference on Shape Modeling Applications (SMI'04)*. 200–208.

Received August 2012; revised April 2013; accepted June 2013