ElasticPaint: A Particle System for Feature Mapping with Minimum Distortion

Christopher Carner and Hong Qin
State University of New York at Stony Brook
{ccarner|qin}@cs.sunysb.edu

Abstract

Mapping of features such as texture and geometric detail is an important tool that enhances realism of surface geometry in graphics and animation. This essentially involves a transformation of 2D coordinates with associated attributes to a target surface in 3D. Moreover, users are often in need of more powerful techniques that can also enable cut-and-paste functionality to directly map a feature from one surface in 3D to another with high-fidelity. Such an operation should minimize feature deformation on the target surface. In practice, it is also desirable to hide geometric complexities and transformations from ordinary users, requiring minimal input and providing an intuitive interface for painting. This paper develops a generalized feature mapping technique using a physically-based particle system to map both geometry and associative attributes between two curved surfaces with the physically-correct minimum distortion of these attributes over the target surface.

1 Introduction and Motivation

Graphical modeling demands efficient design tools to produce high-quality surfaces and realistic images. For example, users can manipulate control points of splines to define and deform curved models. They benefit from additional intuitive operations that make the system more flexible and facilitate model creation and manipulation. Operations that are useful to artistic designers include texture mapping and cut-and-paste functionality. Users demand more powerful techniques that can combine components of different models into a new model smoothly and efficiently. This allows them to embellish their models with new effects while saving the extra time and effort required to manually change the surface to include the desired attributes.

However, existing techniques for painting surface models with new effects fall short of artists' expectations for several reasons. First, previous methods may proceed indirectly, or in several stages. For example, the method by Biermann et al. [2] requires three separate transformations to produce the final mapping; two intermediate transformations are computed to map both the source and target regions to a plane. Next, a linear transformation is used to align the two planar parameterizations. This can be very time consuming, because it involves angle-based flattening of a large connected set of points several times. In addition, planar techniques are only limited to mapping those surfaces with the topology of a disc; a more complicated region may fold over on itself when it is flattened. In general, these methods tend to overly distort the feature due to irregularity in the target surface (e.g., in areas of high curvature). Therefore, the integrity of the feature's physical attributes and photo-realism is not preserved.

By contrast, a physics-based approach can be used to provide a *direct* mapping that overcomes these limitations. Physics-based particle systems are very versatile tools that can serve as the foundation for many applications, providing an automatic, adaptive, and physically intuitive solution. The general formulation of a particle system includes a set of particles, or *"smart points"*, whose behaviors are subject to a set of energy constraints. The interaction of these particles drives the physical system as it evolves over time and ultimately determines their final physical configuration. In addition, particle systems produce a set of intermediate particle configurations that constitute a natural simulation or physical animation.

This paper utilizes a physics-based particle system to provide a systematic mechanism to map not only geometry, but also any associative details from one surface to another. In essence, our system generalizes texture and other feature mapping in two ways: (1) compared with regular texture mapping involving a transformation from a 2D plane to a 3D object, our system allows the user to paint a feature from one 3D object to another one directly; and (2) any attribute, such as color or bump map vector associated with a geometric point on the model can be mapped to the target surface with high accuracy. In particular, our system computes this mapping with the physically-accurate minimum distortion. The particle system is based on simple physical laws, producing a very natural simulation and solution. Its operation is intuitive for naive users, only requiring boundary curves on two surfaces to be interactively specified. This streamlines the system development with a simplified interface, as well as minimal intervention by the user, hence hiding complexities of the geometric transformation while producing a natural feature mapping. An added benefit of our system is that the designer is not restricted to painting on complicated 3D surfaces. In fact, the user can draw on a more simplified region, even a plane, and use our mapping technique to paste the feature onto a convoluted surface with minimum deformation.

2 Research Contribution

We accomplish the goal of least distortion by using a particle system. We combine several energy constraints to provide an automatic and adaptive mapping. We use an energy formulation devised by [7] to create a particle configuration that accurately samples the important feature points on the source surface. We adaptively compute the mapping by enforcing the particles to conform to our formulation of minimum distortion. As a result of combining these energy formulations with the general particle system, we are able to find the physically-accurate minimum-distortion mapping of the feature from the selected region to the destination surface.

The main contributions of this paper consist of:

- a physics-based technique that utilizes energy minimization to find the feature mapping in 3D with minimal user interaction, a natural and intuitive solution, and the *least possible distortion*.
- a *generalized* mapping technique that can add any features associated with the geometry to the new surface.
- a particle system framework that can combine several energy constraints to create an automatic, adaptive solution for a variety of applications.
- a mathematical formulation that quantifies distortion of a texture mapped from a source to target surface.
- a mathematical formulation of an attraction force that magnetizes the surface, allowing physical particles that act according to a combination of forces.

3 Previous Work

Particle systems have proven to be very powerful in many applications. They were used by Reeves [12] in modeling natural "fuzzy" phenomena such as fire and water. Although Reeves' particles did not physically interact, they still reacted to a set of system constraints and produced desirable results. Shimada et al. [13] presented a method for automated mesh generation, showing the versatility of particle systems applied to non-manifold geometry. Szeliski and Tonnesen [14] introduced oriented particles to model surfaces, in which the physical particles constitute the surface without manual connectivity. They developed energy functionals that automatically configured the particles into common surface and solid configurations. Terzopoulos used dynamic models for free-form deformable models [16], [15]. Later, Qin and Terzopoulos [10] combined NURBS with a dynamic formulation to produce D-NURBS, surfaces that allow more intuitive control of their geometry. McDonnell and Qin [8] used physics-based techniques to produce a dynamic free-form deformable modeling system based on subdivision solids.

A point sampling method developed by Li [7] finds a set of feature points in surface areas of high curvature. Li provides an appropriate energy expression for this purpose (also see Section 4). In addition, the system will create a sampling from random initialization of points that preserves topology. However, it is limited to this particular application because of the procedure it uses for solving the energy minimization formula as well as its restriction to mere positional points.

Another sampling scheme by Witkin and Heckbert [18] utilizes a particle system to sample implicit surfaces. This method differs from ours in that the particles are not explicitly connected, but local particle interaction is governed by a repulsion energy distribution and adaptive fissioning based on local density. As a result, neighbors must be determined for every change in particle configuration (i.e. at every frame of the simulation).

A technique for mapping ("pasting") surfaces was introduced by Barghiel et al. [1] that used a hierarchy of splines to produce a layer of composite surfaces. In this system, only a small number of points on the surface were mapped. As a result, their method computes displacement mappings at interactive speed. Chan et al. [3] make this method more flexible by allowing the user to effect changes on the texture or feature not only in the domain, but in the target space as well. Conrad et al. [4] enhance this technique using quasi-interpolation to create an approximated boundary of the feature, which greatly alleviates the discontinuity between the pasted feature and the target surface.

However, these methods do not address the issue of distortion caused by the transformation of the texture domain to the target domain. Biermann's method [2] uses a multiresolution subdivision hierarchy to represent the surfaces. Although this makes it possible to compute the mapping of the texture at interactive speed, their use of planar flattening introduces additional distortion in regions of high curvature. Levy et al. [6] segment the model into charts and use a least squares conformal parameterization method to paint on a target surface while minimizing deformations. On the other hand, our work incorporates physics into the model to minimize the distortion.

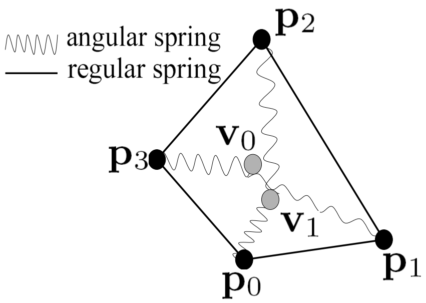

Figure 1. A face in the grid of particles on the surface. Particles are black; virtual points are gray.

4 Particle System Formulation

4.1 Particles and Spring Structure

The particle system introduces a set of physical properties to the surfaces such as mass points, stiffness, damping, and spring constraints. Each particle consists of position, velocity and acceleration vectors, mass, and two sets of neighboring springs. Each spring in the set of "normal" springs corresponds to a neighboring particle in the grid, and it contains a rest length and stiffness coefficient. The rest length is used to store the original spacing, to be sustained after the mapping, between sampled particles on the source surface. Stiffness is used to regulate the force applied by the spring on each of its two neighbors.

While the first set of springs is used to maintain the Euclidean distance between particles, this is not sufficient to ensure a least-distortion mapping. In particular, it does not constrain the grid of particles from shearing. So, a set of angular springs is used to maintain the angle between two vectors on each corner of the grid, reducing local shearing distortions. Each angular spring, like a regular spring, consists of two neighboring positions, a rest length, and a stiffness coefficient. However, while one of these positions is a particle in a grid face, the other corresponds to a "virtual" point, calculated as the midpoint of the two neighbors of the particle in the same grid face. Thus, each particle is connected to four massless points, whose positions change as the locations of the neighboring particles are updated during the simulation (Figure 1). As the rest length of an angular spring changes, a force is applied to the "real" particle in the grid (as opposed to the "virtual" point), attempting to maintain the angle on its corner of the grid. In addition, we apply an opposite force with half the magnitude to each of the real particle's neighbors to conserve the energy of the system. As a result, these springs combined with the normal particle-to-particle constraints retain the geometry of the original region as much as possible, resisting local structural and shearing changes (refer to [8] and [9] for more details on angular springs).

The particle system also consists of several boundary particles, which are introduced when the user selects the region on the source surface. These curve points are used to define the region boundary, and are connected to the outside of the grid of sampled particles. To select a region, the user manipulates a curve drawn in the 2D parametric domain of the source surface. To minimize interaction, we do not involve the user in parameterization choices or details of source and target region alignment. The user need only select the desired curve, and the system will perform a direct, automatic mapping. This implementation allows for a simple, intuitive interface. Before the mapping of the particles inside the curve boundary, the boundary particles are transformed to the new curve using arc length parameterization. This provides the minimum distortion mapping for the boundary. Then, during the interior distortion minimization stage, we fix the curve particle positions to maintain this mapping as well as a stable boundary.

4.2 Particle-driven Least Distortion

When performing a mapping from an undeformed texture space, where the texture to be pasted is defined, to a new surface space, there are infinitely many selections. In essence, we must find the best possible correspondence between points in texture space to points in the destination surface space. Since the source surface may have a largely different parameterization over its domain from the target surface when we pick a mapping, the distortion of the texture pasted onto the target surface may be large. Specifically, we need to find the mapping with minimum distortion of the texture domain over the target surface range. Thus, we must define how to measure distortion. Assume $\mathbf{s}(u, v)$ is the function that defines the undeformed texture domain. The associated detail information is characterized as $c(u, v)$. Mathematically, the goal is to find a mapping $\mathbf{f}(\mathbf{s}(u, v))$= $\mathbf{d}(u', v')$, where $\mathbf{d}(u, v)$ defines the target surface. In particular, we must identify two functions $k(u, v)$ and $l(u, v)$ such that

$$\mathbf{f}(\mathbf{s}(u, v)) = \mathbf{d}(k(u, v), l(u, v)). \tag{1}$$

We utilize differential geometry to define minimum distortion. Consider the small neighborhood around (u, v) in the texture domain and $(k(u, v), l(u, v))$ in the target domain created by slightly perturbing u and v by δu and δv, respectively. If there is zero distortion:

$$\begin{aligned}\|\mathbf{d}(k(u + \delta u, v), l(u + \delta u, v)) - \mathbf{d}(k(u, v), l(u, v))\| \\ = \|\mathbf{s}(u + \delta u, v) - \mathbf{s}(u, v)\|,\end{aligned} \tag{2}$$

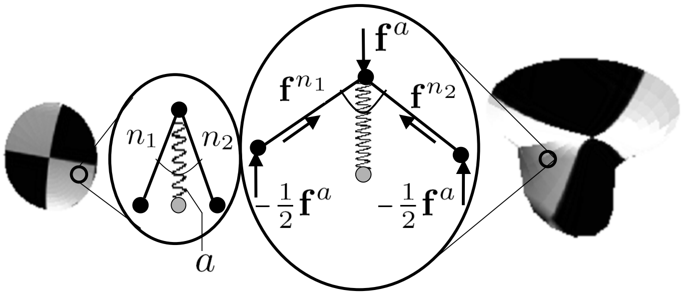

Figure 2. A source neighborhood (left) naively mapped to the target region (right). $\mathbf{f}^{n_1}$ and $\mathbf{f}^{n_2}$ sustain the structure of the geometry, while angular force $\mathbf{f}^a$ maintains the angle between the two regular springs.

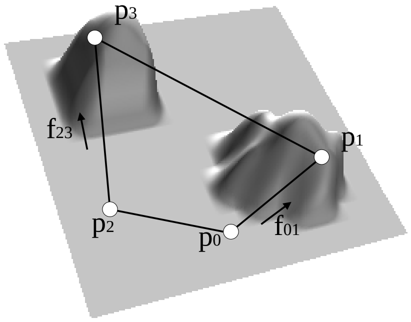

Figure 3. The sampling forces displayed on the source region before the mapping. Particles migrate to highly curved areas during adaptive sampling.

and

$$\|\mathbf{d}(k(u,v+\delta v),l(u,v+\delta v))-\mathbf{d}(k(u,v),l(u,v))\| = \|\mathbf{s}(u,v+\delta v)-\mathbf{s}(u,v)\|. \tag{3}$$

So the distortion equation for change in $\mathbf{u}$ (similar for $\mathbf{v}$) is:

$$\int(\|\mathbf{d}_k(k(u,v),l(u,v))k_u+\mathbf{d}_l(k(u,v),l(u,v))l_u\| - \|\mathbf{s}_u(u,v)\|)^2\,du=0,$$

The discretized relationship (similar for $\mathbf{v}$) is:

$$\sum_u[\|\mathbf{d}(k(u+\triangle u,v),l(u+\triangle u,v))-\mathbf{d}(k(u,v),l(u,v))\| - \|\mathbf{s}(u+\triangle u,v)-\mathbf{s}(u,v)\|]^2=0, \tag{4}$$

We approximate this functional using a set of particles. The vector from particle $\mathbf{p}_i$ to $\mathbf{p}_j$ estimates the small neighborhood around $\mathbf{p}_i$. We let $\mathbf{s}(u_i,v_i)$ be the position of $\mathbf{p}_i$ on the surface. Once the positions on the original surface are set, $\|\mathbf{s}(u_j,v_j)-\mathbf{s}(u_i,v_i)\|$, becomes the constant rest length k_{ij}, which the system attempts to maintain during the physical simulation. We write $\mathbf{w}_i$ for (u_i,v_i). The energy for each $\mathbf{p}_i$ is:

$$\phi_i^l(\mathbf{w}_i)=\sum_{j\epsilon N_i}[\|\mathbf{d}(k(\mathbf{w}_j),l(\mathbf{w}_j))-\mathbf{d}(k(\mathbf{w}_i),l(\mathbf{w}_i))\| - \|\mathbf{s}(\mathbf{w}_j)-\mathbf{s}(\mathbf{w}_i)\|]^2, \tag{5}$$

where N_i is the set of neighbors to particle $\mathbf{p}_i$. See the Appendix for a more detailed derivation.

In addition, the angle (i.e., dot product) between the two pairs of vectors (2) and (3) is the same. This condition is met using angular springs.

4.3 Adaptive Sampling

In order to provide an accurate model of the texture area on the source surface, we use energy minimization to produce a sample set of particles that is dense in areas of high curvature and sparse in areas that are largely flat (see Figure 3). The adaptive sampling technique we use samples the surface correctly from random initialization. Assume that our preliminary configuration of points are parameterized at $\mathbf{w}_i = (u_i,v_i)$ in the domain of the surface. The energy functional that needs to be minimized for this task is [7]:

$$\phi_i^s(\mathbf{w}_i)=\sum_{j\epsilon N_i}r(\mathbf{w}_j)\|\mathbf{w}_i-\mathbf{w}_j\|^2, \tag{6}$$

where $\mathbf{w}_i$ is the position of particle $\mathbf{p}_i$ in the planar domain, and N_i is the set of particles neighboring $\mathbf{p}_i$. The optimal locations of sample points in the domain satisfy the equation (see Appendix):

$$\sum_{j\epsilon N_i}r(\mathbf{w}_j)(\mathbf{w}_i-\mathbf{w}_j)=0, \tag{7}$$

where $r(\mathbf{w}_j)$ is a weighting function sensitive to the shape of the surface at $\mathbf{w}$. Our shape weighting function $r(\mathbf{w}_j)$ is based on the curvature at $\mathbf{w}$. We use:

$$r(\mathbf{w}_j)=\sqrt{\frac{(\kappa_1(\mathbf{w}_j)^2+\kappa_2(\mathbf{w}_j)^2)}{2}}=\kappa(\mathbf{w}_j),$$

where κ_1 and κ_2 are the curvatures in the u and v directions, respectively, and $\kappa(\mathbf{w}_j)$ is the curvature at $\mathbf{w}_j$.

This formulation provides an accurate sampling of the surface. However, it is limited because the points are restricted to the planar (u,v) domain. In order to make this a true physical model, we seek particles in 3D that are not confined to the surface, but can move freely in response to forces applied. This makes the system more flexible because it does not restrict our model to using forces that apply only in the planar domain. Thus, we use an alternative equation to force the particles onto the surface:

$$\sum_{j\epsilon N_i}r(\mathbf{x}_j)\mathbf{T}_i(\mathbf{x}_i-\mathbf{x}_j)=0, \tag{8}$$

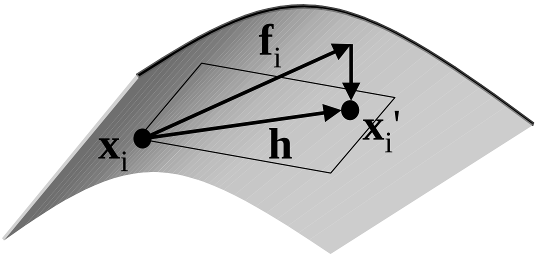

Figure 4. When particle $\mathbf{p}_i$ with position $\mathbf{x}_i$ leaves the surface due to the sum of the internal forces $\mathbf{f}_i$ acting upon it, an additional "magnet" force pulls $\mathbf{p}_i$ back onto the surface.

where $\mathbf{x}_i$ is the particle position $\mathbf{p}_i$ in 3D, and $\mathbf{T}_i$ is the projection of $\mathbf{x}_i - \mathbf{x}_j$ onto the tangent plane of the surface at $\mathbf{x}_i$.

While we realize that this method will not represent the best sampling of the target region, the difference in geometry of the two regions can be alleviated by using a dense sampling rate. In addition, it is also possible to provide a better sampling of the target region by subdividing the grid of particles over the region after the minimum distortion mapping is found.

4.4 Magnetic Surface

In order to obtain an equilibrium of particles that remains on the object, we "magnetize" the surface with another energy constraint, based on the Taylor series expansion:

$$\mathbf{s}(u+\triangle u, v+\triangle v) = \mathbf{s}(u,v) + \mathbf{s}_u(u,v)\triangle u + \mathbf{s}_v(u,v)\triangle v.$$

We project the vector $\mathbf{f}_i$ back onto the tangent plane of the surface at (u,v) to get $\mathbf{h} = proj_{\mathbf{s}_u(u,v)\triangle u+\mathbf{s}_v(u,v)\triangle v}(\mathbf{f}_i)$. Thus, we get a new target position $\mathbf{x}'_i$ on the tangent plane of the surface. If $\mathbf{x}_i = \mathbf{s}(u,v)$, then $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{h}$. So the energy relationship becomes $\phi_i^m(\mathbf{x}_i) = c\|\mathbf{x}'_i - \mathbf{x}_i\|^2$, where c is a spring stiffness coefficient. If any other force, such as the push from a neighboring spring, moves a particle off the surface, this will project the particle back onto the tangent plane of the surface at a position near the point where it lost contact (see Figure 4).

4.5 Dynamics Equations

The physical system will evolve through time subject to the chosen energy minimization constraints. In particular, a particle $\mathbf{p}_i$ is characterized by its position $\mathbf{x}_i$, velocity $\mathbf{v}_i$, and acceleration $\mathbf{a}_i$ at any time t. We need to compute the position $\mathbf{x}_i$ at time $t+1$ from time t. This is governed by the discretized form of Lagrangian equations of motion:

$$m_i\mathbf{x}''_i + \gamma_i\mathbf{x}'_i + \mathbf{f}_i^{int} = \mathbf{f}_i^{ext}, \quad (9)$$

where m_i and γ_i are the mass and damping coefficients, respectively, associated with $\mathbf{p}_i$, $\mathbf{f}_i^{int}$ is the total internal force applied to $\mathbf{p}_i$ by neighboring particles, and $\mathbf{f}_i^{ext}$ is the total external force on $\mathbf{p}_i$. In our system, there are no forces on any particle other than those produced by other internal particles, so $\mathbf{f}_i^{ext} = 0$. Also, $\mathbf{f}_i^{int} = \nabla_{x_i}\phi_i$. This system is flexible because it allows us to define an assortment of mathematical energy relationships, each chosen for a specific application, that govern the physical behavior of the particles.

4.6 Numerical Integration

We use an explicit integration method to solve the system of force balance equations for the new acceleration $\mathbf{a}_i(t+1)$, velocity $\mathbf{v}_i(t+1)$, and finally the new position $\mathbf{x}_i(t+1)$ of particle $\mathbf{p}_i$. Although this requires us to use a small timestep t to ensure stability of the system, the implementation is much simpler, and the procedure is faster than an implicit integration scheme. We compute the total force $\mathbf{f}_i^{int}$ on each particle $\mathbf{p}_i$ and then use the following system to integrate over the time interval t:

$$\mathbf{a}_i(t+1) = \frac{\mathbf{f}_i^{int}}{m_i}, \quad (10)$$

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \mathbf{a}_i(t+1)\triangle t, \quad (11)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)\triangle t. \quad (12)$$

The integration is computed repeatedly as the physical system evolves to a state of equilibrium.

5 Particle System Implementation

Data structures used in this system include a set of linked lists to contain the particles and springs: one for the boundary particles, one for the interior particles, and one for the springs. Each particle has several attributes such as position, velocity, acceleration, mass, a set of four neighboring "normal" springs, and a set of four neighboring "angular" springs. Each spring includes two neighboring particles, a rest length, and a stiffness coefficient. We use a C++ class to include the set of particles and springs, as well as for calculation of the inter-particle forces. Another class is used to handle commands from the GUI to the system, including rendering, region selection, and reading in textures.

The GUI is written with GLUI [11], which provides a useful set of window tools as well as good support for OpenGL. The surfaces are rendered using OpenGL on an Nvidia GeForce2 Pro graphics card. The system was tested on a Windows 2000 PC using a single Intel 1.4 GHz Pentium 4 processor with 512 MB RAM.

Source	Target	Particles	Springs	Time(sec)
Plane	Head	176	1158	3.37
Cylinder	Goblet	322	1986	8.22
Plane	Vase	322	1993	21.26
Plane	Torus	345	2127	14.42
Vase	Torus	409	2551	17.15
Goblet	Torus	198	1258	9.18
Plane	Cylinder	2183	13175	156.42
Goblet	Vase	2037	12363	161.30
Plane	Cap	5775	34757	205.69

Table 1. Mapping statistics for several data sets used in our system. The runtime refers to the total time for convergence to a minimum energy configuration.

Table 1 shows statistics of several mappings of various source and target surfaces. The runtime included in this table refers to the total time for the simulation, which ended when the system reached a stable, minimum energy configuration.

6 Conclusions

We have developed a new method for transferring detail from one surface to a destination surface with minimum distortion of the source texture. Our technique generalizes texture mapping in that we can map not only from a 2D plane, but also from a 3D surface. In addition, rather than just color, we can transfer any type of detail and features associated with the geometry. Our system integrates physics-based principles with curved surface geometry. We combine several energy minimization formulations to achieve the physically accurate least distortion transformation. Physics-based particle systems are automated and adaptive, finding the best solution according to the energy constraint. As a result, our system allows for a clean, simple interface that requires minimal user intervention as well as an intuitive physical solution.

Our system can be enhanced to find the least-distortion mapping not only over NURBS surfaces, but also over arbitrary meshes. We can also create a faster simulation by limiting the number of particles used in the physical formulation. Although this would create a coarser model, we can approximate the solution by using some interpolation or subdivision method. Such a modification would trade a degree of precision for speed. Also, our system presently does not handle mapping sections of NURBS models with seams or discontinuities in the parameterization. However, our implementation can be expanded to include this capability by allowing for additional input along with the NURBS model. By storing and examining the location of the seams, the enhanced system would be able to extend the particle grid to span several charts or patches, if so desired. These issues are currently under investigation.

Acknowledgements

This research was supported in part by the NSF ITR grant IIS-0082035, the NSF grant IIS-0097646, the Alfred P. Sloan Fellowship, and the Honda Initiation Award. We would also like to thank Scott Braut for his texture romanemblem.jpg (Copyright © 2000 TextureUniverse.com and Scott Braut).

References

[1] C. Barghiel, R. Bartels, and D. Forsey. Pasting spline surfaces. *Mathematical Methods for Curves and Surfaces*, pages 31–40, 1994.

[2] H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. *ACM Transactions on Graphics*, pages 312–321, 2002.

[3] L. K. Y. Chan, S. Mann, and R. Bartels. World space surface pasting. *Proceedings of Graphics Interface*, pages 146–154, 1997.

[4] B. Conrad and S. Mann. Better pasting via quasi-interpolation. *Curve and Surface Design*, pages 27–36, 2000.

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Proceedings of SIGGRAPH 1995*, pages 173–182, 1995.

[6] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *Proceedings of SIGGRAPH 2002*, pages 362–371, 2002.

[7] S. Z. Li. Adaptive sampling and mesh generation. *Computer-Aided Design*, 27(3):235–240, 1995.

[8] K. T. McDonnell and H. Qin. Sculpting and animation of free-form subdivision solids. *Proceedings of IEEE Computer Animation 2000*, pages 126–133, 2000.

[9] K. T. McDonnell and H. Qin. Dynamic sculpting and animation of free-form subdivision solids. *The Visual Computer*, pages 81–96, 2002.

[10] H. Qin and D. Terzopoulos. D-nurbs: a physics-based framework for geometric design. *IEEE Transactions on Visualization and Computer Graphics*, pages 85–96, 1996.

[11] P. Rademacher. www.cs.unc.edu/ rademach/glui/.

[12] W. T. Reeves. Particle systems – a technique for modelling a class of fuzzy objects. *Proceedings of SIGGRAPH 1983*, pages 359–376, 1983.

[13] K. Shimada and D. Gossard. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. *ACM Symposium on Solid Modeling and Applications*, pages 409–419, 1995.

[14] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. *Proceedings of SIGGRAPH 1992*, pages 185–194, 1992.

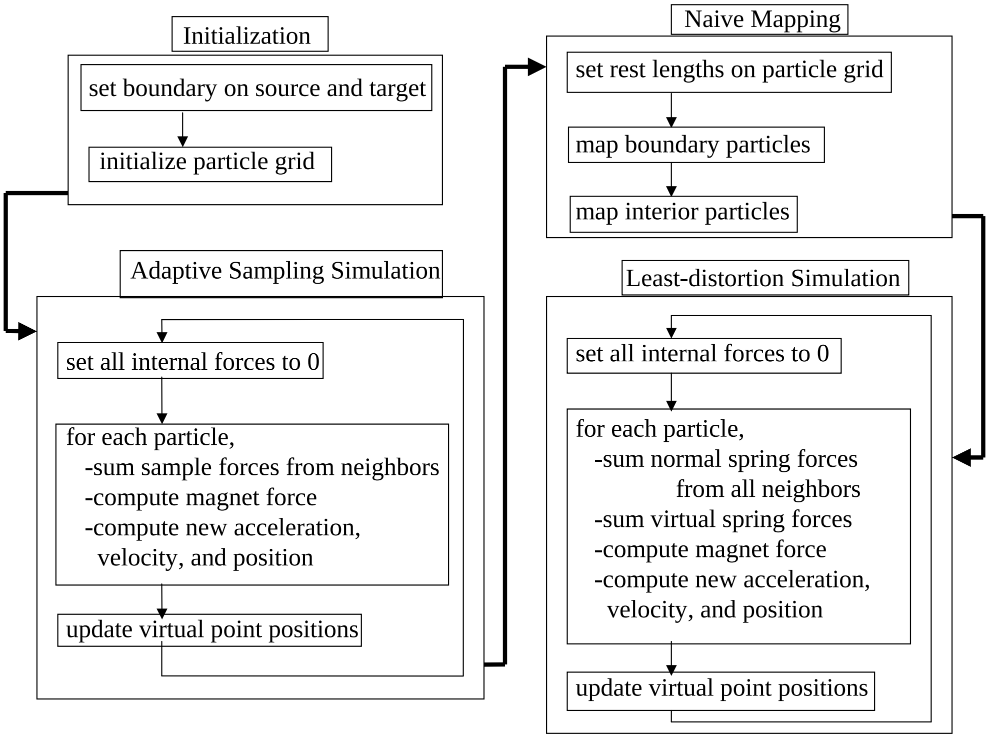
Initialization
set boundary on source and target
initialize particle grid
Naive Mapping
set rest lengths on particle grid
map boundary particles
map interior particles
Adaptive Sampling Simulation
set all internal forces to 0
for each particle,
-sum sample forces from neighbors
-compute magnet force
-compute new acceleration,
velocity, and position
update virtual point positions
Least-distortion Simulation
set all internal forces to 0
for each particle,
-sum normal spring forces
from all neighbors
-sum virtual spring forces
-compute magnet force
-compute new acceleration,
velocity, and position
update virtual point positions

Figure 5. The system architecture.

[15] D. Terzopoulos and K. Fleischer. Deformable models. *The Visual Computer*, pages 306–331, 1988.

[16] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Proceedings of SIGGRAPH 1987*, pages 205–214, 1987.

[17] D. Tonnesen. *Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction, and Animation*. PhD thesis, University of Toronto, 1998.

[18] A. Witkin and P. Heckbert. Using particles to sample and control implicit surfaces. *Proceedings of SIGGRAPH 1994*, pages 269–277, 1994.

Appendix

The energy minimization formulae given in Sec. 4 translate into force relationships. The system evolves over time to a rest configuration with the minimum possible energy allowed by these constraints. We begin with a derivation for the least distortion force. We then sum all the forces applied on each particle $\mathbf{p}_i$ to get the total force. From (4) and the equivalent for change in $\mathbf{v}$ we get:

$$\sum_u \sum_v [\|\mathbf{d}(k(u+\triangle u, v+\triangle v), l(u+\triangle u, v+\triangle v)) - \mathbf{d}(k(u,v), l(u,v))\| - \|\mathbf{s}(u+\triangle u, v+\triangle v) - \mathbf{s}(u,v)\|]^2 = 0.$$

When we sample the surface $\mathbf{s}(u,v)$ with particles, we let $u_i = u$ and substitute u_j for $u + \triangle u$ (similarly for v), and the discretized relationship becomes:

$$\sum_{i\epsilon A}\sum_{j\epsilon N_i}[\|\mathbf{d}(k(u_j,v_j),l(u_j,v_j))-\mathbf{d}(k(u_i,v_i),l(u_i,v_i))\| - \|\mathbf{s}(u_j,v_j)-\mathbf{s}(u_i,v_i)\|]^2 = 0, \tag{13}$$

where A is the set of particles and N_i is the set of neighbors to particle i. We let $k_{ij} = \|\mathbf{s}(u_j,v_j) - \mathbf{s}(u_i,v_i)\|$ be the constant rest length. Also, we let $\mathbf{d}_{ij} = \mathbf{d}(k(u_j,v_j),l(u_j,v_j)) - \mathbf{d}(k(u_i,v_i),l(u_i,v_i))$. Thus, for each particle $\mathbf{p}_i$ in the set of n particles within the boundary curve on the source surface, the energy between particles $\mathbf{p}_i$ and $\mathbf{p}_j$ is $\phi^l_{ij} = c(\|\mathbf{d}_{ij}\| - k_{ij})^2$. Therefore, the corresponding force becomes:

$$\mathbf{f}^l_{ij} = \nabla_{\mathbf{x}_i}\phi^l_{ij} = C\frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|}(\|\mathbf{d}_{ij}\| - k_{ij}). \tag{14}$$

where $\mathbf{x}_i = \mathbf{d}(k(u_i,v_i),l(u_i,v_i))$ is the position of particle $\mathbf{p}_i$ and C is a constant stiffness coefficient. For the sampling energy, we have $\phi^s_{ij} = c(r(\mathbf{x}_j))\|\mathbf{x}_i - \mathbf{x}_j\|^2$. Thus, sampling force upon particle $\mathbf{p}_i$ due to particle $\mathbf{p}_j$ becomes $\mathbf{f}^s_{ij} = C(r(\mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j) + r'(\mathbf{x}_j)\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. However, as long as the sampling rate is sufficiently high, the neighborhood around $\mathbf{x}_i$ is relatively flat, meaning $r'(\mathbf{x}_j) \approx 0$. So we have:

$$\mathbf{f}^s_{ij} = Cr(\mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j).$$

Finally, we have the magnetic surface energy, which keeps particles on the surface: $\phi^m_i = c\|\mathbf{x}'_i - \mathbf{x}_i\|^2$. Thus, the corresponding force on particle $\mathbf{p}_i$ becomes:

$$\mathbf{f}^m_i = C(\mathbf{x}'_i - \mathbf{x}_i).$$

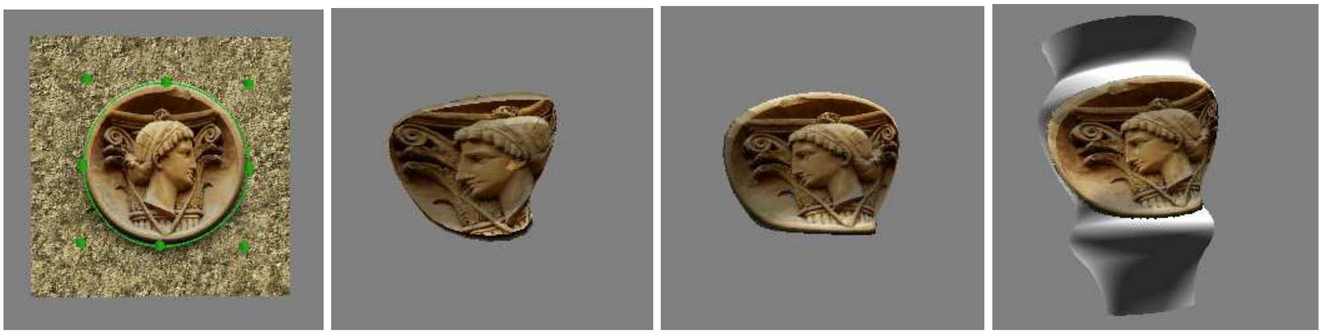

Figure 6. (a) Roman emblem to be mapped. (b) A distorted mapping. Notice that the stretching in certain areas can be alleviated (e.g. the nose). (c) Emblem after 326 time steps. (d) Least-distorted mapping.

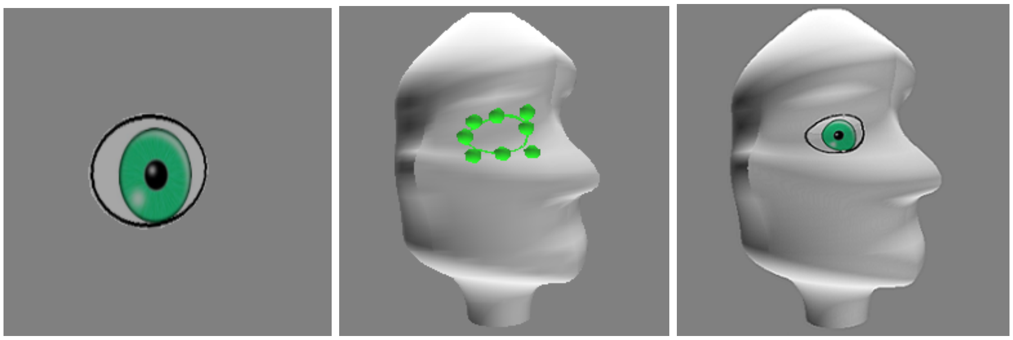

Figure 7. Mapping an eye to a NURBS head.

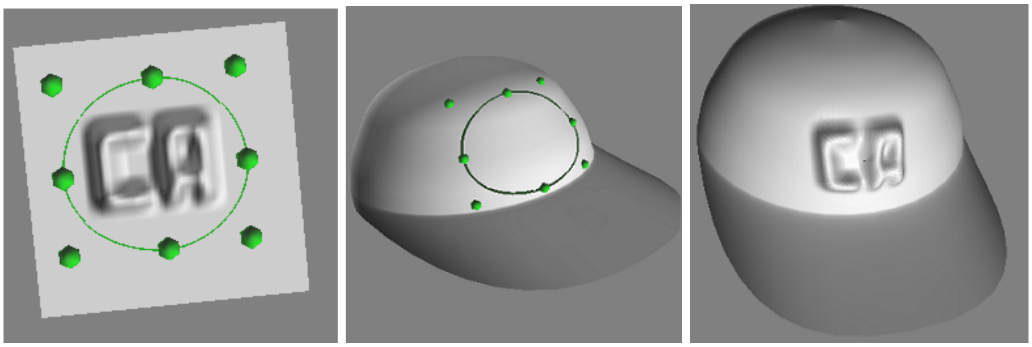

Figure 8. Instead of color, we associate a height with each point in 3D space.

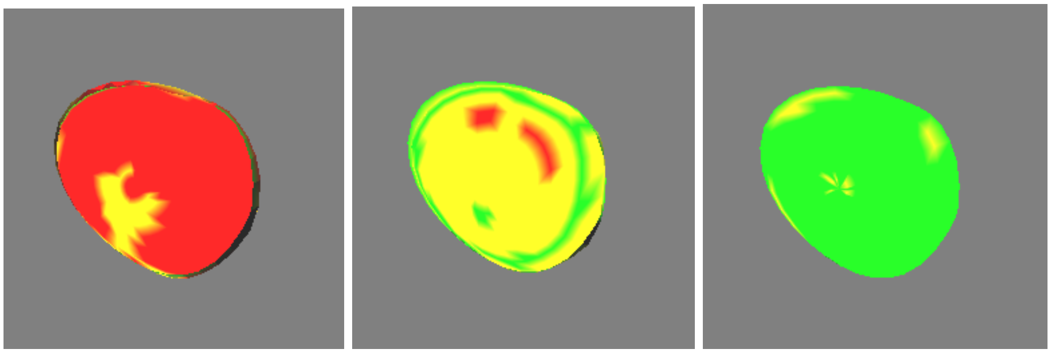

Figure 9. (a) A naive mapping. Red, yellow, and green indicate sections of high, medium, and low distortion, respectively. (b) Distortion after 40 time steps. (c) Distortion after 92 time steps.