Robust Particle Systems for Curvature Dependent Sampling of Implicit Surfaces

Miriah D. Meyer
School of Computing
University of Utah
miriah@sci.utah.edu

Pierre Georgel
Department of Computer Science
École Normale Supérieure Paris
georgel@clipper.ens.fr

Ross T. Whitaker
School of Computing
University of Utah
whitaker@sci.utah.edu

Abstract

Recent research on point-based surface representations suggests that point sets may be a viable alternative to parametric surface representations in applications where the topological constraints of a parameterization are unwieldy or inefficient. Particle systems offer a mechanism for controlling point samples and distributing them according to needs of the application.Furthermore, particle systems can serve as a surface representation in their own right, or to augment implicit functions, allowing for both efficient rendering and control of implicit function parameters. The state of the art in surface sampling particle systems, however, presents some shortcomings. First, most of these systems have many parameters that interact with some complexity, making it difficult for users to tune the system to meet specific requirements. Furthermore, these systems do not lend themselves to spatially adaptive sampling schemes, which are essential for efficient, accurate representations of complex surfaces. In this paper we present a new class of energy functions for distributing particles on implicit surfaces and a corresponding set of numerical techniques. These techniques provide stable, scalable, efficient, and controllable mechanisms for distributing particles that sample implicit surfaces within a locally adaptive framework.

1. Introduction

Implicit surfaces are used for surface modeling [2], point set reconstruction [7], animation [5], and visualization [22]. The widespread use of implicit surfaces is due to several desirable features of the surface representation, such as a simple, homogeneous representation (e.g. volumes), the ability to model deformations without reparameterization, and flexible topology. However, these advantages come at a cost. Implicit surfaces are inefficient to render directly and unintuitive to control.

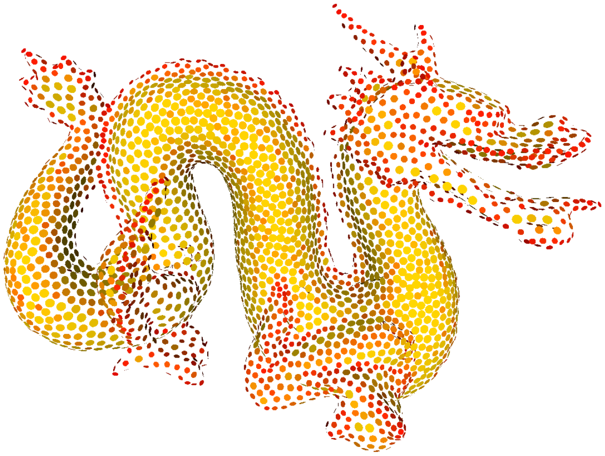


Figure 1. A curvature dependent sampling of the dragon (Figure 5 in the color plate).

In 1994, Witkin and Heckbert [23] introduced a novel approach to sampling and controlling implicit surfaces in which they constrain a system of interacting particles to lie on the implicit surface. Each particle repels nearby particles to minimize a Gaussian energy function. The Gaussian energy has a characteristic length, which is adapted for each particle to suit the distribution of its neighbors. Simultaneously the particles are constrained and reprojected onto the implicit surface. The Witkin and Heckbert (W-H) method includes approximately 10 free parameters, and when they are carefully tuned ([23] gives an effective set of guidelines), the resulting system produces a homogeneous distribution of particles on the surface. Many applications, however, require inhomogeneous distributions based on curvature, such as using particles to polygonalize [11] or to parametrize [24] the surface. Several authors [16, 3] have attempted to enhance the W-H method for locally adaptive sampling , but have noted the difficulty of tuning such a complicated system of parameters.

The difficulty with the W-H method is twofold. The first is the presence of the characteristic length in the inter-particle distances σ, which is the standard deviation of the Gaussian. The Gaussian energy has a preferred value, and particles that are not situated at a distance of approximately σ move very slowly relative to those that are. To account for this, the W-H method adapts the σ on a per particle basis according to a set of heuristics that involve several free parameters. The σ adaptation is intimately tied to an insertion and deletion of particles when σ becomes too large or too small, respectively, and thus the equilibrium requires a particular number of particles, which is determined by the system, rather than the user. The second issue is the numerical algorithm. The W-H method relies on a gradient descent for both particle repulsion and σ adaptation. Discretized gradient descent algorithms invariably introduce a critical free parameter (the descent rate, or unit change per iteration), and the system can easily become either too slow or unstable if the parameter is improperly tuned. As a result, changes to the W-H system often entail careful retuning of the corresponding gradient descent parameter.

This paper addresses both of these issues. The effects of the characteristic length are alleviated by introducing an energy function that is (approximately) scale invariant. Thus, particles interact in a similar fashion over a wide range of distances without adapting or tuning parameters. To address the limitations of gradient descent we propose a inverse-Hessian minimization scheme, which automatically tunes the descent rate to accommodate the curvature of the energy surface. The result is a robust system with relatively few parameters that provides a new capability: a locally adaptive distribution of particles on implicit surfaces. Mechanisms such as neighborhood size and deletion/insertion of particles can now be adapted to meet other constraints, such as the total number of particles in the system, the average particle density, the efficiency of the computation, or update and rendering times.

2. Related Work

Modeling surfaces with particles was first proposed in computer graphics by Szeliski and Tonnesen [18, 19], by developing an oriented particle system to sample deformable surface models. They employ an energy function from the molecular dynamics literature, which causes particles to exert short-range repulsion and long-range attraction, keeping particles at an appropriate distance from each other. Turk [21] uses repelling particles to re-sample polygonalized static surfaces using curvature measurements, while De Figueiredo *et al.* [4] proposes a physically-based particle method to polygonalize implicit surfaces by modeling particles with a mass-spring system. Building on these ideas, Witkin and Heckbert [23] propose a novel, physically-based system that uses repulsive particles to not only sample, but also deform, implicit surfaces. Other work presents ideas for sampling implicit surfaces for animation [6] and texture mapping [24]. Also, a large body of work has been developed in the mathematics community that studies the discretization of surfaces via energy minimizations [17, 8].

Heckbert [10] extends the original W-H method by developing a spatial binning optimization that determines the radius of influence of a particle and only calculates forces for neighboring particles within this radius. The radius of influence varies from particle to particle, so the bounding sphere must be computed for each particle. Another extension is proposed by Hart *et al.* [9] to sample more complex surfaces. They describe an object-oriented system that numerically differentiates implicit surfaces comprised of large numbers of control parameters. Rösch *et al.* [16] introduce a curvature dependency into the W-H method for sampling unbounded surfaces and singularities. Similarly, Crossno *et al.* [3] propose modifying the W-H scheme to accommodate local curvature of extracted isosurfaces. Research in point-based graphics, [1, 14, 7] builds, in part, upon ideas from particle systems. Point-based methods fit an implicit surface to a point cloud and then utilize particle system techniques to refine the point representation.

3. The Witkin-Heckbert Method

The W-H method constrains n repulsive particles $\boldsymbol{p}(t)$ to lie on an implicit surface of the function $F(q)$, which is controlled by parameters $\boldsymbol{q}(t)$. That is,

$$F(\boldsymbol{p}(t), \boldsymbol{q}(t)) = 0. \tag{1}$$

In [23], particles provide not only a way to visualize the implicit surface in real time, but also provide a handle through which surface deformations are controlled by updating the parameters $\boldsymbol{q}(t)$. This paper is concerned with only the particle placement, and thus, to simplify the discussion, we ignore the surface deformation terms from the original W-H formulation.

Particles distribute themselves homogeneously across the surface by exerting an inter-particle repulsion force on neighboring particles, resulting in a repulsion velocity $\boldsymbol{v}_i$ for each particle. The surface constraint is enforced by projecting $\boldsymbol{v}_i$ onto the local tangent plane and then reprojecting the updated particle position onto the zero set of F using a Newton-Rhapson technique. That is,

$$\dot{\boldsymbol{p}}_i = \left(I - \frac{\nabla F_i \otimes \nabla F_i}{\nabla F_i \cdot \nabla F_i} \right) \boldsymbol{v}_i - \phi F_i \frac{\nabla_i F}{\nabla F_i \cdot \nabla F_i}, \tag{2}$$

where I is the identity matrix, ∇F_i is the spatial gradient of F at the particle $\boldsymbol{p}_i$, and $\nabla F_i \otimes \nabla F_i$ is the projection operator formed by the vector direct product of the gradient.

The last term in (2) is the reprojection onto the implicit surface (the *feedback term* in [23]), which is scaled by a free parameter ϕ. A fraction of $\dot{\boldsymbol{p}}_i$ is added to the current position in the manner of a finite forward difference scheme, i.e. $\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i + c\dot{\boldsymbol{p}}_i$, where c is the gradient descent constant mentioned previously. The particle is then rendered using a disk oriented to lie in the local surface tangent plane [18].

Each particle maintains an adaptive repulsion radius, σ_i, which grows and shrinks based on the local energy values. The W-H method also includes a target radius, $\hat{\sigma}$, that controls the insertion and deletion of particles in the system. When a particle's σ_i drops below some fraction of $\hat{\sigma}$ it is removed, and when it goes above some multiple of $\hat{\sigma}$, a new particle is inserted nearby. The system can quickly move particles into sparse regions by growing σ_i for particles in underpopulated regions, and then inserting new particles when σ_i becomes too high. The system forms homogeneous distributions of particles over the surface by adapting σ_i until all particles have similar energy measures.

Several extension to the original W-H particle system have been proposed in order to accommodate increased sampling in areas of high curvature [16, 3, 11]. All three extensions apply an adaptive, per-particle, curvature dependency to either the repulsion radius, σ_i, or to the target radius, $\hat{\sigma}_i$. We have found these extensions do not provide adequate curvature dependent distributions for complex surfaces with large variations in curvature values, confirming the difficulties mentioned by Rösch *et al.* [16] and Karkanis *et al.* [11].

These difficulties arise because σ_i will grow and shrink regardless of the underlying curvature value, and $\hat{\sigma}_i$ does not control the behavior of a particle apart from its splitting and dying. For example, consider a particle in a high curvature area with a relatively low $\hat{\sigma}_i$. As σ_i increases this particle splits at faster rates than particles in nearby flatter areas. However, these new particles will merely be pushed out onto flat areas, which will, in turn, become too crowded, resulting in the deletion of particles. Meanwhile, the high-curvature particle, missing the fleeing particles it recently created, will continue to split—a never ending cycle of insertion and deletion. In our experiments, when we tuned parameters to stop the insertion-deletion cycle by expanding the hysteresis of insertion and deletion we found that the particle distributions did not reflect the desired differences in particle densities—the W-H scheme tends toward homogeneous distributions despite variations in $\hat{\sigma}$. Local adaptation in the W-H scheme is significantly more complex than a single parameter; it also requires modifications to the particle radii interactions with the per-particle energy functions.

This paper proposes a new approach to distributing particles across an implicit surface, allowing for a wide range of distribution patterns, from homogeneous to highly adaptive. The proposed system is general across a broad range of shape complexity and size, and requires minimal parameter tuning from surface to surface. We build upon the constrained particle system developed by Witkin and Heckbert, but introduce a new class of energy functions accompanied by a single, global radius σ that virtually eliminates the need for insertion-deletion to ensure even distributions of particles across the entire surface. We apply particle insertion and deletion to add global particle density control, and create a curvature-based repulsion amplitude parameter that causes particles to distribute with higher densities in regions of interest.

4. A New Particle Energy Scheme

At the heart of the proposed particle system is the computation of the potential energy associated with particle-particle interactions. The minimization of this energy defines the algorithm for distributing particles across the surface and leads to a quantifiable notion of an ideal distribution. Each particle, $\boldsymbol{p}_i$, creates a potential field, which is a function of the distance $|\boldsymbol{p}_i - \boldsymbol{p}_j|$ between $\boldsymbol{p}_i$ and all neighboring particles that lie within the potential field. We define the energy at a particle $\boldsymbol{p}_i$ to be the sum of potentials of the m particles that interact with $\boldsymbol{p}_i$:

$$E_i = \frac{1}{2} \sum_{j=1}^{m, j \neq i} E_{ij}(\mid \boldsymbol{r}_{ij} \mid), \tag{3}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{p}_i - \boldsymbol{p}_j$. The global energy of the system is the sum of all the individual particle energies. Because the pairwise energies, E_{ij}, are symmetric we have

$$E = \sum_{j=1}^{m} \sum_{i=j+1}^{n} E_{ij}(\mid \boldsymbol{r}_{ij} \mid). \tag{4}$$

The derivative of the system energy with respect to a particle position gives rise to the repulsive force that defines the desired velocity direction (steepest descent):

$$\boldsymbol{v}_i = -\frac{\partial E}{\partial \boldsymbol{x}_i} = -\frac{\partial E_i}{\partial \boldsymbol{x}_i} = -\sum_{j=1}^{m, j \neq i} \frac{\partial E_{ij}}{\partial \mid \boldsymbol{r}_{ij} \mid} \frac{\boldsymbol{r}_{ij}}{\mid \boldsymbol{r}_{ij} \mid}. \tag{5}$$

By iteratively moving the particles along the energy gradient, we find a global minimum when particles are evenly distributed across the surface.

The pairwise potential energy E_{ij} is the most important aspect of any such particle system. A bad energy function can lead to numerical instabilities and uneven particle distributions, and a good function results in a homogeneous steady state. We have experimented with several potential energy functions from the literature and have identified three important characteristics of a well behaved potential energy function. Energies should be continuous functions of particle distance. The energy functions should be compact to avoid global influences and allow for efficient computation. And to avoid characteristic lengths, and the associated parameter tuning, the energy must be scale invariant. That is, two particles at different distances should have the same ratio of energies regardless of the choice of units of our system.

The Gaussian energy used in the W-H method is smooth and nearly compact because the function can be truncated in a manner that does not significantly affect its behavior. However, the Gaussian has a characteristic length and is not scale invariant. A particularly interesting example of a scale-invariant energy is the electrostatic potential, $E_{ij} = 1/|\boldsymbol{r}_{ij}|$. The electrostatic function is smooth, except at the origin (which can be fixed by adding adding a small constant to the denominator), but does not fall off quickly enough to provide local behavior. As a result, particles do not remain on flat regions but instead concentrate exclusively on convex, high-curvature areas—a well-known phenomenon from electrostatics. Furthermore, truncating the electrostatic potential yields unreliable results, and the configurations of particle steady states are very sensitive to the distance of truncation. Thus, the electrostatic function is not approximately compact. Figures 2a-b show graphs of the Gaussian and electrostatic energy functions, respectively.

A particularly desirable energy function, which establishes a good compromise between approximate scale invariance and compactness, is a modified cotangent:

$$E_{ij} = \begin{cases} \cot\left(\frac{|\boldsymbol{r}_{ij}|}{\sigma}\frac{\pi}{2}\right) + \frac{|\boldsymbol{r}_{ij}|}{\sigma}\frac{\pi}{2} - \frac{\pi}{2} & |\boldsymbol{r}_{ij}| \leq \sigma \\ 0 & |\boldsymbol{r}_{ij}| > \sigma \end{cases} \qquad (6)$$

which is shown graphically in Figure 2c. This potential has one free parameter σ, which establishes the farthest distance at which particles interact. The derivative of this energy with respect to particle distance is:

$$\frac{\partial E_{ij}}{\partial \mid \boldsymbol{r}_{ij} \mid} = \begin{cases} \frac{\pi}{2\sigma}\left[1 - \sin^{-2}\left(\frac{|\boldsymbol{r}_{ij}|}{\sigma}\frac{\pi}{2}\right)\right] & |\boldsymbol{r}_{ij}| \leq \sigma \\ 0 & |\boldsymbol{r}_{ij}| > \sigma \end{cases} \qquad (7)$$

The derivative shows an analogous relationship to the electrostatic potential. When the distance between particles is small relative to σ, $\sin(|\boldsymbol{r}_{ij}|/\sigma) \approx |\boldsymbol{r}_{ij}|/\sigma$, and the force behaves like $-1/r^2$, which is invariant to scale.

Our experiments show that the cotangent energy homogeneously distributes particles across the surface, freeing up the need to modify σ on a per particle basis or to implement a parameter sensitive particle insertion and deletion algorithm. The particles distribute themselves in a nearly hexagonal packing, which is the general pattern for optimal configurations [17]. The system is well behaved due to the lack of a characteristic length in the energy or force function, and works across a broad range of surface shapes and sizes with no modifications. Because of this robust behavior, σ can be treated as an application dependent parame-

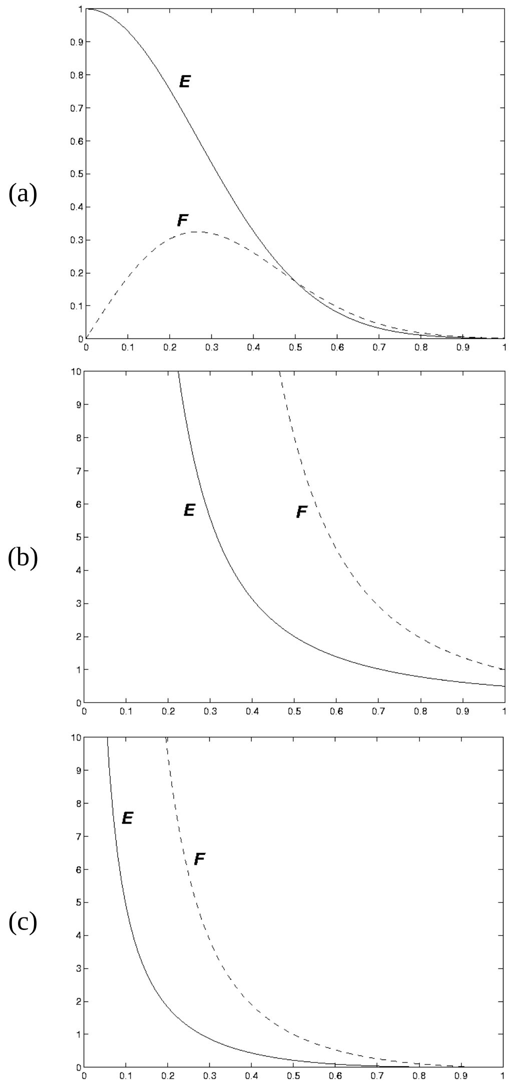


Figure 2. Plots of energy functions (E) and the corresponding force functions (F) : (a) Gaussian energy, exhibiting a characteristic length; (b) Electrostatic energy, exhibiting a necessary truncation; (c) Cotangent energy, exhibiting compactness and approximate scale invariance.

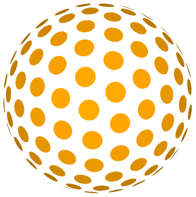
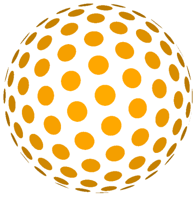
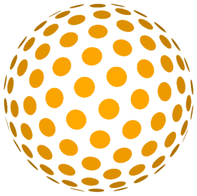
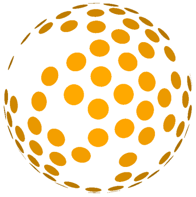
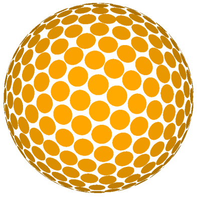
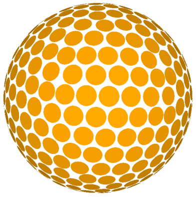
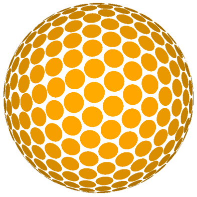
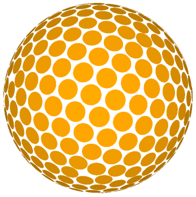
n	$\sigma = 1$	m	$\sigma = \frac{1}{6}$	m	$\sigma = \frac{1}{9}$	m	$\sigma = \frac{1}{12}$	m
300		150.3		8.5		5.8		1.1
600		302.1		18.9		6.6		5.8

Table 1. The effects of varying the one free parameter, σ, in the cotangent energy function. The total number of particles, n, is constant along the rows, and the value of σ is the fraction of the total domain. The average number of influenced neighbors, m, is given for each variation. The upper right example illustrates the one constraint on σ – σ must be large enough to ensure adequate distributing forces at a particle.

ter which can be tuned in accordance with the desired density of particles and the run-time requirements of the application, as we will discuss in Section 6.

Table 1 illustrates the robustness and generality of the cotangent energy function. For these examples we start with a random placement of particles, then iteratively move these particles using a simple gradient descent until the system converges to a homogeneous distribution. We vary the value of σ under two different scenarios: a sparsely packed system of 300 particles, and a densely packed system of 600 particles. When $\sigma = 1$ (where σ is the fraction of the domain size), the energy has a global influence over the entire surface function domain. The particle distributions continue to be homogeneous as we reduce σ, demonstrating that the cotangent energy is approximately invariant over a wide range of scales. The only restriction is that σ must be large enough so that particles interact with a ring of neighbors at the steady state ($m \approx 6$ for this example). The upper right example in Table 1 shows how the system breaks down when σ is too small to provide sufficient interaction. As discussed in later sections, this condition can be met automatically by either increasing σ or adding more particles.

5. Moving Particles

Integrating the particles towards a progressively lower energy state can introduce numerical challenges. A gradient descent requires careful tuning of the step size parameter, which can vary from surface to surface, and even particle to particle. Improper values can lead to very long convergence times if the step size is too small, or irreconcilable oscillations around the minimum if the step size is too large. To avoid these problems we have implemented a Levenberg-Marquardt integration scheme that does not require any tuning of parameters (described in the following subsection).

We propose a Gauss-Siedel update strategy, in which particle positions are updated one at a time, and each particle update relies on the most recent positions of its neighbors. Moving particles in this method, versus finding all particle movements before making positional updates, allows us to avoid large matrix operations for global minimization, and instead works locally on small matrices using a global metric to define convergence.

A two-step update scheme keeps the moving particles on the surface. First, particle positions are updated based on the repulsion velocity in the tangent plane:

$$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i + \left(I - \frac{\nabla F_i \otimes \nabla F_i}{\nabla F_i \cdot \nabla F_i} \right) \boldsymbol{v}_i, \qquad (8)$$

which is the result of a Lagrangian formulation of the constrained optimization that keeps particles on the zero sets of F. However, movements in the tangent plane can push particles off of the surface, especially in areas of high curvature. Therefore one must follow up with a reprojection:

$$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i - F_i \frac{\nabla F_i}{\nabla F_i \cdot \nabla F_i}, \qquad (9)$$

which is a Newton-Rhapson approximation to the nearby roots (zero sets) of F.

Levenberg-Marquardt

Positioning particles across the surface is a nonlinear optimization problem, and the inverse-Hessian techniques, such as the Levenberg-Marquardt algorithm, are often very effective at producing stable results in a timely manner [15]. Levenberg-Marquardt (L-M) works by varying between an inverse-Hessian when the energy is locally quadratic or a gradient descent when the inverse-Hessian fails. We begin with an expression for the Hessian of the particle potential. The Hessian of a particle is a 3×3 matrix composed of the summed second derivatives of the energy with respect to a particle's position:

$$\begin{aligned} H_i &= \frac{\partial^2 E_i}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_i} \\ &= \sum_{j=1}^{m} \left[\frac{\partial^2 E_{ij}}{\partial |\boldsymbol{r}_{ij}|^2} (\boldsymbol{n}_{ij} \otimes \boldsymbol{n}_{ij}) \right. \\ &\qquad \left. + \frac{1}{\mid \boldsymbol{r}_{ij} \mid} \frac{\partial E_{ij}}{\partial \mid \boldsymbol{r}_{ij} \mid} (I - \boldsymbol{n}_{ij} \otimes \boldsymbol{n}_{ij}) \right] \qquad (10) \end{aligned}$$

where $\boldsymbol{n}_{ij}$ is the normalized $\boldsymbol{r}_{ij}$ vector. When computing a particle's Hessian in the L-M algorithm, we ignore the second term of (10), which corresponds to the second derivatives of $\boldsymbol{x}_i$, because inclusion of these second derivative terms can often be destabilizing [15]. This is justified numerically by noting that the sum of first derivatives for any one particle must be zero as it approaches equilibrium.

The L-M method adaptively conditions the Hessian by increasing the diagonal matrix elements with a factor λ:

$$\tilde{H}_{ab} = \begin{cases} H_{aa}(1+\lambda) & \\ H_{ab} & a \neq b \end{cases} \qquad (11)$$

The desired velocity due to tangential repulsive forces is then determined by solving the system

$$\boldsymbol{v}_i = -\left(\tilde{H}_i\right)^{-1} \boldsymbol{D}_i, \text{ where } \boldsymbol{D}_i = \frac{\partial E_{ij}}{\partial |\boldsymbol{r}_{ij}|} \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}. \qquad (12)$$

When λ is large, $\tilde{H}$ becomes diagonally dominate and a particle moves in small steps in order to not over step the minimum, but when λ is small, the particle moves much faster, using a full quadratic approximation to the energy surface.

Each particle maintains a λ value, which is initialized to 1.0 (the system is insensitive to this value) and modified by factors of 10 as the system converges to an even distribution. Modifying λ_i is an iterative process governed by energy computations and comparisons, and works as follows for one particle's position update:

Step 1: Compute $\boldsymbol{D}_i$ and E_i.

Step 2: Compute $\tilde{H}_i$, solve (12), and compute the new energy value, E_i^{new}, at the new particle location $\boldsymbol{p}_i^{\text{new}}$.

Step 3: If $E_i^{\text{new}} \geq E_i$, increase λ_i by a factor of 10 and go back to Step 2.

Step 4: If $E_i^{\text{new}} < E_i$, update $\boldsymbol{p}_i = \boldsymbol{p}_i^{\text{new}}$ and $E_i = E_i^{\text{new}}$, (decrease λ_i by a factor of 10 if this is the first time through the loop,) and move onto the next particle.

We have found that particles must be on the surface before computing E_i^{new} because a particle should not be allowed to move to a lower energy position if that position is not on the surface. We put a particle back on the surface after solving (12) using the reprojection given in (9). Also, the system is slow to converge if the particles are allowed to jump over one another, and thus we penalize a particle with a very high energy when it attempts to move a distance farther than its nearest neighbor. The nearest neighbor penalty forces λ to increase until the motion is on the same scale as the neighborhood. Each particle is updated and moved individually so that changes are propagated into subsequent particle updates. The entire process over all of the particles is repeated until convergence, which is indicated in the L-M algorithm by a very large average λ.

The L-M method requires significantly fewer system iterations to reach a desirable distribution of particles than gradient descent, but each iteration takes longer to compute because of the need to invert the 3×3 Hessian matrix. The most important gain of the L-M method is the insensitivity to the only two free parameters, the initial and final λ values. We have found the L-M algorithm to converge over a wide range of surface shapes and sizes with no modifications.

6. System Control

The scheme described in the previous section ensures that particles repel each other and reach a uniform distribution in a reasonable amount of time, without modifying free parameters, or inserting and deleting particles. The only restriction of the scheme is σ must be large enough such that the given number of particles covers the surface. In practice, when dealing with unfamiliar or deforming surfaces it may be necessary to enforce certain relationships between the particles and the surface. For instance, we might want to maintain a minimum number of particles, a certain minimum particle density, or a minimum particle radius that covers the surface with specified number of particles. Meeting these conditions will require modifying the number of particles or the radius of the energy function. Furthermore, we would also like to have particle distributions that adapt to local surface features.

6.1. Global Control of Particles

As with the W-H method, the particle energy quantifies the density of particles in the neighborhood of a single particle, while the system energy provides information on the global density of particles. The system energy measure provides further insight to the efficiency of the system and the locality of the particle influences. Based on energy measures, several techniques can be used to ensure an efficient and effective system.

The energy of a particle quantifies the amount of interaction the particle has with its neighbors, where a low energy implies too few neighbors, and high energy indicates to many. To determine whether the particle system contains enough energy to enforce even particle distributions without incurring unnecessary particle-particle computations, the system energy measure is compared against an ideal energy measure, E_{ideal}. In our system, we define E_{ideal} to be a hexagonal packing of particles, similar to the distribution described in the W-H method. The hexagonal configuration represents a natural, low local energy distribution [17], and is illustrated in Figure 3. When the system energy is below E_{ideal}, particles will generally not have enough neighbors to reach an acceptable distribution. Conversely, when the system energy is greater than E_{ideal}, particles are influencing more neighbors than necessary, resulting in extraneous particle-particle computations and slower global convergence.

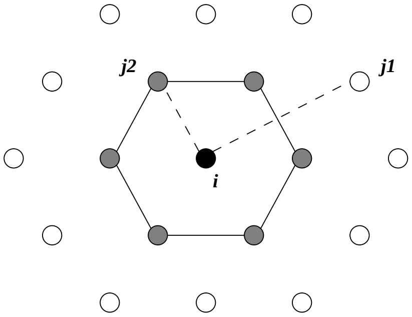


Figure 3. When determining the ideal energy at a particle, we want only the 6-ring neighbors to influence a particle's energy and force calculations. In this diagram, we want the distance from i to $j1$ to be σ, which makes the distance from i to $j2$ approximately 0.57σ. The shade of the particles specifies the relative energy and force influence to the center particle.

To achieve the E_{ideal}, several mechanisms exist to modify the system energy. Particles can be inserted or deleted to increase or decrease the system energy respectively, or σ can grow or shrink. These mechanisms can be used separately or in combination, depending on the goals of the application.

Inserting and deleting particles drives the system to maintain a specific surface density of particles, defined by the value of σ. For a specific σ value, the system energy specifies the approximate local density of neighboring particles across the entire surface. If the local densities contain more particles than the defined ideal packing, particles can be deleted, either randomly across the surface, or in a biased approach similar to the W-H deletion criteria. Conversely, low local energies can be adjusted by splitting particles in the local tangent plane.

Adjusting σ can be used when a specific number of particles is desired. The σ value grows and shrinks to ensure that particles interact with only the ideal neighborhood distribution. Growing σ is important when the system energy is too low to ensure an even distribution of particles, while shrinking σ when the system energy is too high produces the most computationally efficient system by keeping inter-particle calculations as local as possible. Changes in σ are carried out iteratively using either a gradient descent, inverse-Hessian, or some other one-dimensional optimization technique.

For our system, we utilize a combination of insertion and deletion of particles, and growing and shrinking σ to maintain a lower bound on the number of particles and an upper bound on σ. This combination of constraints ensures a minimum number of particles in the system at all times, and a cap on the complexity of the inter-particle calculations.

6.2. Locally Adaptive Particle Distribution

The local density of particles can be controlled to achieve an adaptive sampling by introducing a repulsion amplitude, α_i, for each particle. The symmetric form of the pair-wise potential functions associated with this modification is:

$$E_i = \sum_{j=1}^{m} \frac{(\alpha_i + \alpha_j)}{2} E_{ij}. \tag{13}$$

The repulsion amplitude parameter terms are then propagated into the force and Hessian equations, (5) and (10).

Applications that entail complex implicit surfaces benefit from increased sampling density near high-curvature features. To accomplish this curvature dependent sampling within the proposed framework, α_i is scaled by the inverse of the local curvature norm, κ_i, at $\boldsymbol{p}_i$. In order to keep a minimum distribution in flat regions and to prevent surface creases from attracting too many particles, we put a soft minimum and maximum on this scaling:

$$\alpha_i = \gamma + \frac{1}{\kappa_i^d + \nu}. \tag{14}$$

The upper bound on α_i is $1/\nu$, and the lower bound is γ. We find that $\nu = 0.75$, $\gamma = 0.1$, $d = 2$ and a curvature range of $[0, 20]$ works well and distributes particles with a strong curvature dependence. To increase the curvature dependence, d can be increase in (14), or to smooth out the dependency, $d < 1$ can be used. We calculate the curvature of the implicit surface from the derivatives of F using the method described by Kindlmann *et al.* [12]. By adding α_i into the energy, force, and Hessian equations, the influence of particles in regions of high curvature is reduced, causing particles to form denser configurations near these surface features.

7. Implementation and Results

In our implementation of the proposed curvature based sampling method we eliminate divisions by zero by adding a very small value to all denominators. For all of the results presented in this paper we initialized our system with the number of particles the system is required to maintain, in random positions within the 3D domain of the implicit surface, and the σ value is set as a fraction of the domain. The system then iterates until the particle distribution converges by repeating the following steps:

1. For each particle:
   (a) Compute $\boldsymbol{v}_i$ and $\hat{H}_i$.
   (b) Find E_i and the new particle position using the two-step particle motion and reprojection (Section 5), combined with the L-M algorithm (Section 5) for computing velocities. Because of the finite region of influence of the energy function, we compute pairwise relationships for only those particles that are within the radius σ. For efficiency we use a spatial binning structure as described by Crossno *et. al* [3].
   (c) Update the particle's position and check whether the particle has moved to a new spatial bin.
2. Decide whether the system is at a steady state by checking that the average $\log_{10}(\lambda_i)$ is greater than λ_{max} (for our system we use $\lambda_{max} = 14$). If the system is still moving, go back to the particle update in Step 1.
3. If the system is not moving, check whether we have a desirable configuration by comparing the system energy to $\hat{E}$ using a relative difference threshold. If we do not have an acceptable energy insert or delete particles, or change σ, to meet the application-specific system requirements described in Section 6, and then go back to the particle updates in Step 1 and continue iterating.
4. If the energy is acceptable, stop iterating.

The three examples in Figure 4 (and Figure 7 in the color plate) are generated as the zero set of the quartic function:

$$F = \boldsymbol{x}^4 - 10r^2\boldsymbol{x}^2 \qquad (15)$$

with $r = 0.13$, and centered in the domain $[0, 1]^3$. All three examples are initialized with 1000 particles and $\sigma = 0.125$, and required to maintain the initial number of particles. The left image is generated by setting $\alpha_i = 1$. The middle and right images use (14) with $d = 1$ and $d = 3$ respectively. On a 1.7GHz P4 processor with 1GB of memory, all three examples converge in 15 to 18 seconds.

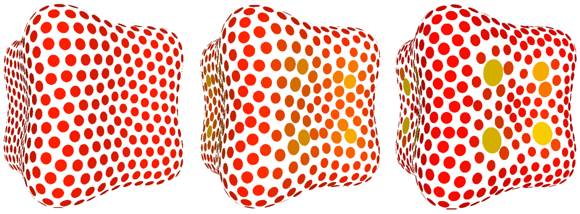


Figure 4. A quartic function with varying equations for the repulsion amplitude, α (Figure 7 in the color plate).

Figure 1, Figure 5 and Figure 6 show several other examples of our system. The torus and box are represented as zero sets of scalar trivariate B-splines, and each contain approximately 1250 particles. The dragon and griffin contain approximately 7000 particles each, and are represented as distance fields. The particles have been colored and scaled according to the repulsion amplitude parameter, α_i, of each particle. The results illustrate the method for clustering particles around regions of high curvature.

8. Conclusions and Future Work

In this paper we have presented a new particle system for robust, adaptive sampling of complex implicit surfaces. We have developed a new class of energy functions and applied several numerical techniques to generalize, stabilize, and control the distribution of particles. The physically-based nature of the particle system inherently incurs a high computational cost, and our system provides the tools necessary to automate the particle convergence reliably.

The feature-based sampling of the particle system allows for more numerically sophisticated techniques to be applied to implicit models. Deformations of the underlying implicit function can be made more efficient and effective by concentrating computations and functional control in regions of interest. The potential energy of the particles can also be used to optimize the sampling of polynomials across the surface [20]. Coupling the particle system with a point based surface splatter, finite element data can benefit from the nature of the surface constrained particle system to improve the accuracy and efficiency of visualizing higher order surfaces [13].

Acknowledgments

The authors wish to thank David Breen of Drexel University and the Stanford 3D Scanning Repository for the dragon and griffin distance fields. We also wish to thank the anonymous reviewers for their constructive comments. This research was supported by the NSF grant CCR-0310705 and the ARO grant DAAD19-01-10013.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, January 2003.

[2] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256, 1982.

[3] P. Crossno and E. Angel. Isosurface extraction using particle systems. In *Proceedings of the 8th conference on Visualization '97*, pages 495–498. IEEE Computer Society Press, 1997.

[4] L. H. de Figueiredo, J. de Miranda Gomes, D. Terzopoulos, and L. Velho. Physically-based methods for polygonization of implicit surfaces. In *Proceedings of the conference on Graphics interface '92*, pages 250–257. Morgan Kaufmann Publishers Inc., 1992.

[5] M. Desbrun and M.-P. Cani. Animating soft substances with implicit surfaces. In R. Cook, editor, *SIGGRAPH 95 Conference Proceedings*, volume 29 of *Annual Conference Series*, pages 287–290. ACM SIGGRAPH, Addison Wesley, aug 1995.

[6] M. Desbrun, N. Tsingos, and M.-P. Cani. Adaptive sampling of implicit surfaces for interactive modeling and animation. In *Implicit Surfaces'95*, avril 1995. Published under the name Marie-Paule Gascuel.

[7] X. Guo and H. Qin. Dynamic sculpting and deformation of point set surfaces. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 123. IEEE Computer Society, 2003.

[8] D. P. Hardin and E. B. Saff. Discretizing manifolds via minimum energy points. *Notices of the American Mathematical Society*, 51(10):1186–1194, November 2004.

[9] J. C. Hart, E. Bachta, W. Jarosz, and T. Fleury. Using particles to sample and control more complex implicit surfaces. In *Proceedings of the Shape Modeling International 2002 (SMI'02)*, page 129. IEEE Computer Society, 2002.

[10] P. Heckbert. Fast surface particle repulsion. In *SIGGRAPH '97, New Frontiers in Modeling and Texturing Course*, pages 95–114. ACM Press, August 1997.

[11] T. Karkanis and A. J. Stewart. Curvature-dependent triangulation of implicit surfaces. *IEEE Computer Graphics and Applications*, 22(2):60–69, March 2001.

[12] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, October 2003.

[13] B. Nelson and M. Kirby. Ray-tracing polymorphic multidomain spectral/*hp* elements for isosurface rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):114–125, 2006.

[14] M. Pauly, R. Keiser, L. Kobbelt, and M. H. Gross. Shape modeling with point-sampled geometry. *ACM Trans. Graph.*, 22(3):641–650, 2003.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[16] A. Rösch, M. Ruhl, and D. Saupe. Interactive visualization of implicit surfaces with singularities. *Eurographics Computer Graphics Forum*, 16(5):295–306, 1996.

[17] E. B. Saff and A. B. J. Kuijlaars. Distributing many points on a sphere. *Mathematical Intelligencer*, 19(1):5–11, 1997.

[18] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 185–194. ACM Press, 1992.

[19] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 140–152, New York, NY, June 1993.

[20] M. A. Taylor, B. A. Wingate, and R. E. Vincent. An algorithm for computing fekete points in the triangle. *SIAM J. Numer. Anal.*, 38(5):1707–1720, 2000.

[21] G. Turk. Re-tiling polygonal surfaces. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 55–64. ACM Press, 1992.

[22] R. Whitaker, D. Breen, K. Museth, and N. Soni. Segmentation of biological volume datasets using a level set framework. In *Proceedings of the Joint IEEE TCVG and Eurographics Workshop*, pages 249–263, 2001.

[23] A. P. Witkin and P. S. Heckbert. Using particles to sample and control implicit surfaces. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 269–277. ACM Press, 1994.

[24] R. Zonenschein, J. Gomes, L. Velho, and L. H. de Figueiredo. Controlling texture mapping onto implicit surfaces with particle systems. In *Proceedings of Implicit Surfaces '98*, pages 131–138, June 1998.

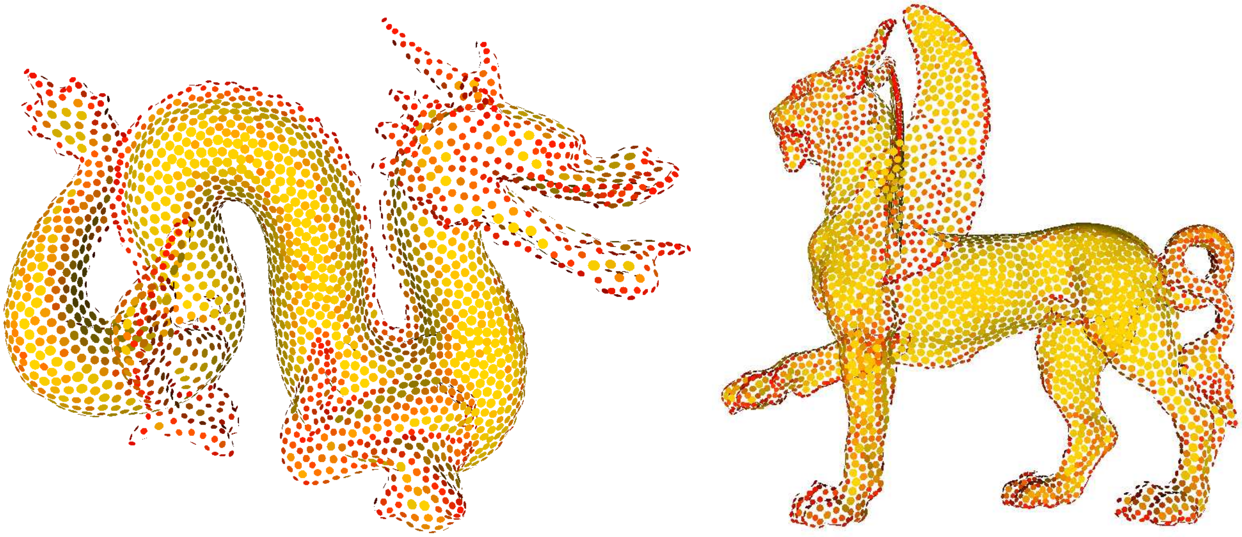


Figure 5. A curvature dependent sampling of the dragon and griffin distance fields.

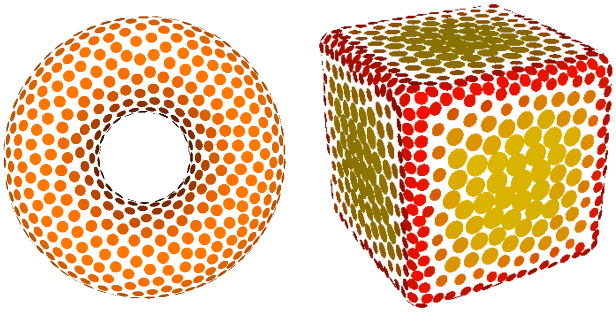


Figure 6. A curvature dependent sampling of the zero sets of scalar trivariate B-splines.

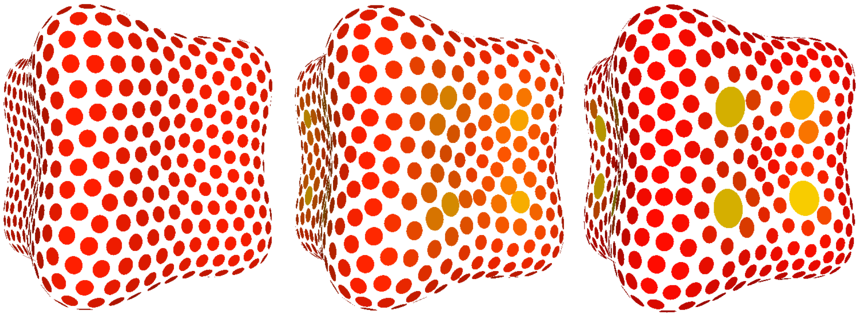


Figure 7. A quartic function with varying equations for the repulsion amplitude, α.