HAL
open science

Fast sampling of implicit surfaces by particle systems

Floriant Levet, Xavier Granier, Christophe Schlick

To cite this version:

Floriant Levet, Xavier Granier, Christophe Schlick. Fast sampling of implicit surfaces by particle systems. SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006, Jun 2006, Matsushima, Japan. pp.39, 10.1109/SMI.2006.13 . inria-00106853v1

HAL Id: inria-00106853

https://inria.hal.science/inria-00106853v1

Submitted on 16 Oct 2006 (v1), last revised 27 Oct 2013 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

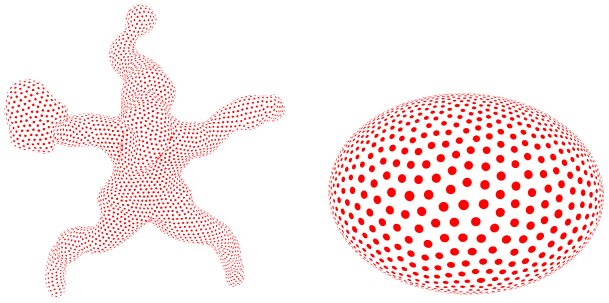
Fast sampling of implicit surfaces by particle systems

Florian Levet and Xavier Granier and Christophe Schlick
IPARLA project - (LaBRI - INRIA futurs)
University Bordeaux 1
Email: [levet,granier,schlick]@labri.fr

Abstract

Particle systems, as originally proposed by Witkin and Heckbert [17], are a powerful way to sample implicit surfaces since they generate almost evenly distributed samples over the surface, thanks to a global minimization of an energy criterion. Nonetheless, due to the computational cost of the relaxation process, the sampling process becomes rather expensive when the number of samples exceeds a few thousands.

In this paper, we propose a technique that only relies on a pure geometry processing which enables us to rapidly generate the set of final particles (e.g., half a second to generate 5,000 particles for an analytic implicit surface) with near-optimal positions. Because of its characteristics, the technique does not need the usual split-and-death criterion anymore and only about ten relaxation steps are necessary to get a high quality sampling. Either uniform or non-uniform sampling can be performed with our technique.



(a) With uniform particles. (b) With non-uniform particles.

Figure 1. Sampled models with either uniform or non-uniform particles.

1 Introduction

In 1994, Witkin and Heckbert [17] gathered existing ideas [13, 14, 2, 15] in a tool designed to both sample and control implicit surfaces. Starting from a seed particle, the surface is sampled thanks to a repulsion scheme and a split-and-death criterion. Each particle repels nearby particles to minimize a Gaussian energy function. The final result is a set of uniform particles almost evenly distributed on the surface.

Some following works have improved this basic scheme by achieving either a non-uniform isotropic sampling [1, 11], an anisotropic sampling [7] or by changing the Gaussian energy function with forces [10] or with new classes of energy functions [9].

Nonetheless, none of these works have focused on the two main drawbacks of particle systems: the computational cost of the algorithm and the convergence of the system. When determining the next position of a particle, one has to compute its intersections with all its neighboring particles. As this process has to be done for each particle, sampling a surface with more than a few thousand particles is too expensive for practical applications. Besides, finding a good convergence criterion that works for all configurations is not an easy task.

The goal of this paper is to enable the use of particle systems with a larger number of particles (more than a few thousands). The basic idea is to rapidly generate the set of final particles with near optimal positions by using pure geometry processing. Therefore, few relaxation steps (around ten) are needed to get a high quality sampling and to avoid the convergence problem. Either uniform or non-uniform sampling can be performed with our technique.

The remainder of the paper is organized as follows: Section 2 presents some related previous work on sampling implicit surfaces by particle systems. Section 3 details our generation algorithm of uniform particles extended in Section 4 to non-uniform particles. Section 5 presents experimental results obtained with our sampling method. Section 6 concludes and presents some directions we are currently investigating.

2 Previous work

Particles were used for the first time to sample a surface in a complete modeling tool based on oriented particle systems developed by Szeliski and Tonnesen [13], but in their work, there was no underlying implicit surface. Turk [14] used a similar process in order to generate textures using a reaction-diffusion method. Figueiredo et al. introduced [2] a system to sample implicit surfaces using particles. It is based on a relaxation process similar to the one proposed by Turk in order to achieve a uniform distribution of the samples. Turk's reaction-diffusion method has also been used to re-tile polygonal surfaces [15] according to the curvature.

Witkin and Heckbert [17] (W-H) developed a powerful modeling application by putting together some of these existing ideas. They defined an adaptive repulsion and a split-and-death condition so that particles could either split or be removed from the surface. Hart et al. [4] improved upon this particle system for automatic and numerical differentiation of implicit surfaces. One main limitation of both works is that only uniform distribution can be generated.

Crossno and Angel [1] derived another extension based on the work of W-H in order to sample an isosurface extracted from a 3D density image. The same repulsion forces and relaxation process is used, but the repulsion radius of each particle is estimated according to the curvature of the sample.

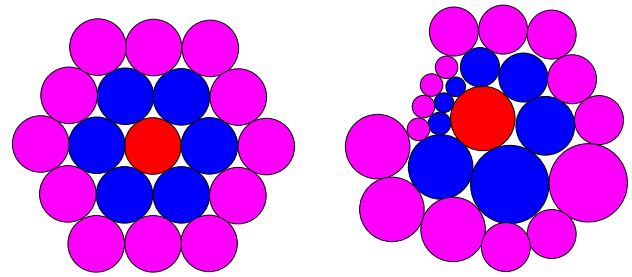
Similarly Rosch et al. [11] extended the W-H scheme to take into account curvature in order to sample unbounded surfaces and singularities. Levet et al. [7] presented an anisotropic sampling of implicit surfaces for differential point rendering [5] by locally taking into account the direction and value of principal curvatures.

Meyer et al. [9] introduced an interesting new class of energy functions for distributing either uniform or non-uniform particles on implicit surfaces. Unfortunately, even if their technique requires less relaxation steps than the original W-H method, each step takes longer.

Pauly et al. [10] proposed to use particle systems in order to simplify point-sampled surfaces (thanks to a death criterion). In their method the distribution of points depends on the curvature of a moving least square (MLS) surface that approximates the set of points [8].

Other works have presented ideas for sampling implicit surfaces for animation [3] and texture mapping [18]. Some particle systems have also been used to polygonize implicit surfaces [2] or as a comparative quality technique for polygonizing implicit surfaces [6].

Finally Su and Hart [12] have presented an object-oriented particle framework designed to rapidly create new particle systems and applications.



(a) Ideal configuration for a uniform sampling.

(b) Possible configuration for a non-uniform sampling.

Figure 2. Configuration examples for the two samplings.

3 Uniform sampling

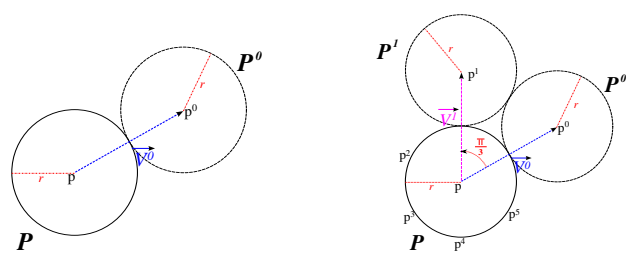
3.1 General approach

For a uniform sampling, the ideal local distribution of particles is given in Figure 2(a). Each particle has six neighbors which is the ideal configuration in order to minimize the energy. Because we work with 3D models, this ideal 2D configuration is hard to achieve. Thus the key idea of our method is to get an efficient generation of particles by a local 2D process, working on tangent planes defined locally for each particle.

3.2 Generation of a particle's children

Let S be the implicit surface we want to sample. Each particle $\mathbf{P}$ is defined by a 3D position p, a normal $\overrightarrow{N}$ at the surface and a repulsion radius r.

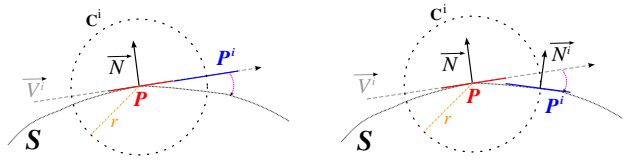
Starting from a seed particle, our goal is to uniformly place its six children particles with respect to its repulsion



(a) The first particle with an arbitrary direction.

(b) The second child is placed at an angle of π / 3.

Figure 3. Generation of particles in the 2D tangent plane.



(a) Initial position of the child $\mathbf{P^i}$.

(b) $\mathbf{P^i}$ is reprojected according to the circle C^i.

Figure 4. Re-projecting a particle on the surface.

radius on the tangent plane of the particle $\mathbf{P}$ defined by its normal $\overrightarrow{N}$.

For each child, we define the tangent vector $\overrightarrow{V^i} = p^i - p$ so as the six particles are regularly placed at an angle of $\pi/3$ with its previous and successive particle as can be seen in Figure 3(b). Their 2D positions are given by:

$$p_x^i = \cos\left(\frac{i\pi}{3}\right)V_x^0 - \sin\left(\frac{i\pi}{3}\right)V_y^0$$
$$p_y^i = \sin\left(\frac{i\pi}{3}\right)V_x^0 + \cos\left(\frac{i\pi}{3}\right)V_y^0$$

Note that the first child of the seed particle is arbitrary initialized (see Figure 3(a)).

3.3 Projecting the children on the surface

The children particles created on the tangent plane of the parent particle are more or less deviated from S depending on the local curvature (see Figure 4(a)). The most common technique used to re-project a particle on the surface is the iterative Newton-Raphson method. But it does not guarantee that the distance between a particle and its children will be preserved.

Instead, we propose to move each child $\mathbf{P^i}$ iteratively on the circle $C^i(p, 2r)$ (see Figure 4(b)) which is the circle centered on p with a radius $2r$, defined on the plane $(p, \overrightarrow{N}, \overrightarrow{V^i})$.

3.4 The complete process

3.4.1 Gathering neighboring particles

With the exception of the seed of the system, we never generate six children. For each particle, some of its children have already been placed by previous particles and exist in the particle set. Thus, before adding the new particles to the set, we have to gather its neighboring particles. We use a space partitioning scheme in order to accelerate this gathering (as in [17, 7]). Any other hierarchical partitioning should work fine.

3.4.2 The algorithm

Starting from an arbitrary chosen seed particle, we generate six children as explained in Section 3.2. These children are added to the stack of untreated particles. This is the initial state of the algorithm.

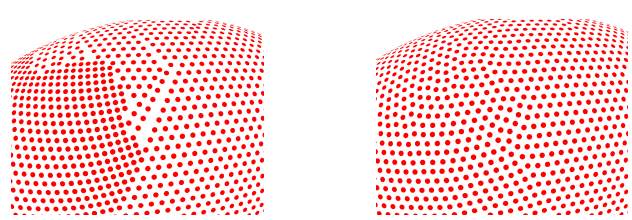
For each particle of the stack, we have to generate six children. We already know a particle at a desired position: its parent. We have to place the five remaining children (that we re-project on the surface). In order not to place children too close to existing particles, we check if they overlap a particle of the neighborhood. If an existing particle corresponds, it replaces the child and this particle is removed from the neighborhood. If there is no particle, the child is added to the particle set and to the stack of untreated particles.

3.5 The relaxation process

The previous step results in a set of particles with positions near the desired ones where each particle has six neighbors. We have a near-optimal sampling in region of low curvature but holes may appear in regions of high curvature (see Figure 5(a)). This comes from the re-projection of the particles from the tangent plane to the surface. A few relaxation steps of the W-H scheme [17] reduce this problem (see Figure 5(b). Based on a Gaussian computation of the energy, it is easy to implement and works well for uniform sampling. We just skip from our implementation the adaptive repulsion radius (as we have particles everywhere on the surface) and the split-and-death criterion (as we already have the desired number of particles).

4 Non-uniform sampling

The main difference with the uniform case is that the repulsion radius is different for each particle and is curvature-dependent. The distance between two particles is no more



(a) After the generation holes can appear.

(b) After the relaxation all holes have disappeared.

Figure 5. Relaxation applied on a model.

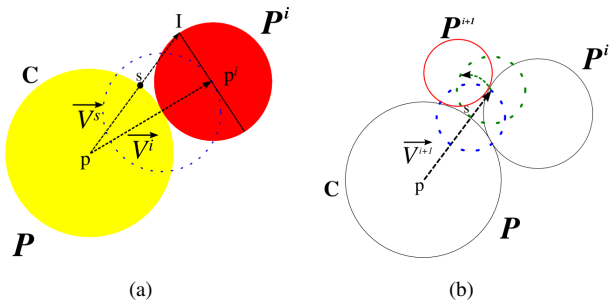


Figure 6. Placement of two children.

$2r$ but instead $r^i + r^j$ (an example of non-uniform sampling is shown in Figure 2(b)). The curvature, computed with the method given in [7], is $\kappa = (\kappa_1 + \kappa_2)/2$ (κ_1 and κ_2 being the values of the principal directions of curvature). κ is then scaled to be used as the repulsion radius.

Since we can not predict the number of children of a particle anymore, we can not place them directly. Note that we still work on the tangent planes at the surface in order to minimize overlapping between successive children.

For the generation of the first child $\mathbf{P^i}$ of the seed point $\mathbf{P}$, an arbitrary initial position is chosen on the tangent circle $C(p, r)$ at the surface (dot circle in Figure 6(a)). At this time, $\mathbf{P}$ and $\mathbf{P^i}$ intersect each other. Thus, we have to iteratively move $\mathbf{P^i}$ in the direction of vector $\overrightarrow{V^i} = p^i - p$ until they do not overlap anymore as can be seen in Figure 6(a) (at each iteration the repulsion radius r^i is re-evaluated in order to match any change in the surface curvature).

The initial position s of the next child $\mathbf{P^{i+1}}$ is given by the intersection between $\overrightarrow{V^s} = I - p$ and C (see Figure 6(a)). Since this initial position is on C, we have to realize a first repulsion as explained above (see Figure 6(b), $\mathbf{P^{i+1}}$ moves from the blue dot circle to the green dot circle position). But, since two successive children ($\mathbf{P^i}$ and $\mathbf{P^{i+1}}$) can intersect each other, we iteratively move $\mathbf{P^{i+1}}$ tangentially to C until they do not overlap anymore. If another child exist in the motion direction and stop it, we minimize the overlapping between the three particles. The particle $\mathbf{P^{i+1}}$ is discarded when this overlapping is too large.

We want to place new particles on each free space between existing neighboring particles. Figure 7-left shows the gathering of neighboring particles (red circles) around the current particle (black circle). Thus we have just to apply the generation process explained above on each of these free spaces. Figure 7-right shows the result of this algorithm on a particle $\mathbf{P}$ with some particles added (blue particles) while others were discarded (transparent particles).

Finally, at the end of the complete generation, we apply a relaxation scheme based on a computation of forces [15, 10, 7] which is easy to implement and was sufficient to validate our approach.

5 Results and discussion

The prototype system was written in the C++ language. All example models shown in this paper were built with the prototype system by using a PC with 3 GHz processor with 1 GB of main memory (note that we do not use the complete available memory).

The sphere (see Figure 9) and ellipsoidal (see Figure 1(b)) models shown in this paper were analytic implicit surfaces while the bunny (see Figure 8) and the Santa (see Figure 1(a)) models were reconstructed using RBF [16] with a set of 200 initial points (note that the evaluation of an implicit surface reconstructed using RBF takes much more time than for analytic implicit surfaces). For the bunny we removed some points on the ears. Thus a part of the implicit surface was reconstructed between these two ears as can be seen in Figure 8-left. So we increased the genius of the initial bunny model.

Since our objective is to quickly generate particles on implicit surfaces, we present some times related to our techniques. Table 1 shows times taken by the sampling of two models: the ellipsoid and the bunny. We compare our uniform sampling with the Witkin-Heckbert technique [17] and our non-uniform sampling with a scheme using only forces [15, 10, 7]. A different number of particles and last relaxation number of particles means that the targeted number of particles was not achieved. For example, for the sampling of the bunny with the Witkin-Heckbert method,

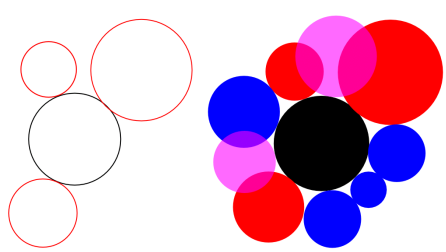


Figure 7. Children generation of a particle.

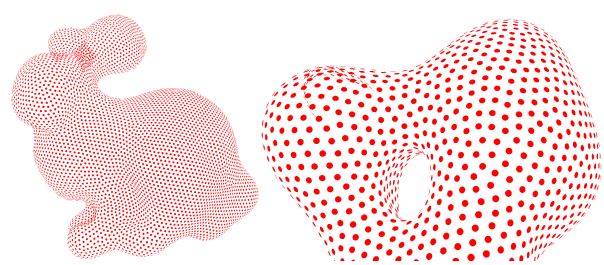


Figure 8. Uniform sampling of the bunny.

Uniform Sampling									
	Our method				Witkin-Heckbert method				
Model	# particles	Last relaxation		Total time	# particles	# relaxation	Last relaxation		Total time
		# particles	Time				# particles	Time	
Ellipsoid	13,923	13,923	1.86	19.80	10,633	373	10,633	1.44	219.63
Ellipsoid	55,748	55,748	59.05	613.54	$\simeq$ 50,000	1,613	33,343	17.79	4,344.01
Bunny	47,356	47,356	151.55	1,691.94	$\simeq$ 50,000	1,959	33,483	68.13	32,912.02
Non-Uniform Sampling									
	Our method				Forces scheme				
Model	# particles	Last relaxation		Total time	# particles	# relaxation	Last relaxation		Total time
		# particles	Time				# particles	Time	
Ellipsoid	27,314	27,314	26.41	252.19	27,151	2,136	27,151	27.04	4,535.15
Bunny	12,388	12,388	16,79	313.77	$\simeq$ 13,000	1,124	10,570	13.88	2,294.73

Table 1. We compare our uniform sampling with the Witkin-Heckbert method and our non-uniform sampling with a scheme based on forces. All given times are in seconds. Only ten relaxation iterations were applied with our techniques. For the two comparative techniques, we start from one seed particle. We give the last relaxation number of particles and time. Contrary to our techniques for which the number of particles remains constant during the relaxation process, this number keeps increasing for each new relaxation step achieved for the comparative techniques.

the targeted number was around 50,000 particles. Because the time needed by the sampling was already too long, we stopped the process before its end when only a part of the surface was sampled (there was only 33,483 particles after 1,959 relaxations).

As said before, our techniques do not need a split criterion (so we keep the number of particles constant during the ten relaxation steps). On the contrary, the two comparative techniques start from one seed particle. Then, the systems are supposed to sample the totality of the surface thanks to their split criterion. The split criterion used for the Witkin-Heckbert method is given in [17]. For the forces scheme, we split a particle when no force is applied on it. Note that we used the same code for the relaxations of our methods and for the comparative techniques (with the exception of the split criterion).

The most interesting observation is that we improve the sampling time on all the examples, enabling the use of particle systems with a larger number of particles (more than a few thousands). Contrary to other systems that need hundreds of relaxation, the use of only ten relaxations saves a lot of time.

Finally, we can see that the non-uniform sampling is slower than the uniform sampling. Since for the non-uniform sampling two iterative repulsions have been introduced, the time needed to generate the children of a particle is larger than for the uniform sampling. Besides we have to compute the curvature of the surface for each particle.

In Table 2 we show the times taken by different levels of sampling of the analytic ellipsoid. We can see that the set of particles generation time is far smaller than the time taken by the relaxation process.

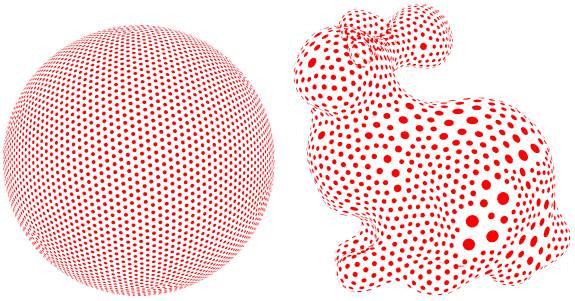


Figure 9. Sampled models with either uniform or non-uniform particles.

Even if the times taken by our methods are very interesting we think that it's possible to still improve them. We use a space partitioning scheme [17, 7] that works very well for a low number of particles but that is far less efficient with

#particles	Time (in second)		
	Generation	Relaxation	Total
2,225	0.11	0.49	0.60
13,923	1.22	18.58	19.80
55,748	24.80	588.74	613.54
87,043	61.60	1,487.49	1,549.09

Table 2. Times taken by the sampling of the analytic ellipsoid.

thousands particles. When a lot of particles are generated, the time of traversal of the structure keeps increasing. Thus, with a better spatial distribution as in [1, 9], we believe that we could speed-up the generation time.

6 Conclusion and future works

In this paper, we have presented a new particles generation algorithm enabling the use of particle systems with a larger number of particles (more than a few thousands). We did not limit our technique to uniform particles since non-uniform distribution of particles can be achieved.

Because our computations only rely on pure geometric processing (by the use of tangent planes), we have developed an efficient and rapid generation algorithm. Its result is a set with the desired number of particles with near optimal positions.

Thanks to the characteristics of our generation algorithm, the relaxation process is reduced to a few iterations (around ten) to get a high quality sampling. This is a great improvement compared to existing particle systems that can need hundred steps of relaxation to achieve their sampling.

Because we only work on tangent planes, holes can appear on the surface and, in some areas, particles can overlap themselves. We believe that our sampling quality can be improved by taking into account the local curvature information during the generation of children.

At this moment, the generation time of our non-uniform technique is larger than for our uniform technique because we have introduced two iterative repulsions that need more computation than a direct placement. We are investigating a way to directly place the children even for the non-uniform sampling.

Finally, the time of one relaxation remains problematic. Even if we limit this problem by needing only ten relaxation steps, finding a criterion to identify particles that do not need to move could greatly improve the speed of the relaxation process.

References

[1] P. Crossno and E. Angel. Isosurface extraction using particle systems. In R. Yagel and H. Hagen, editors, *IEEE Visualization '97*, pages 495–498, 1997.

[2] L. H. de Figueiredo, J. Gomes, D. Terzopoulos, and L. Velho. Physically-based methods for polygonization of implicit surfaces. In *Proceedings of Graphics Interface '92*, pages 250–257. CIPS, 1992.

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

[4] J. C. Hart, E. Bachta, W. Jarosz, and T. Fleury. Using particles to sample and control more complex implicit surfaces. In *Proceedings of Shape Modeling International 2002*, pages 129–136, 2002.

[5] A. Kalaiah and A. Varshney. Differential point rendering. In K. Myszkowski and S. Gortler, editors, *Rendering Techniques 2001 (Proceedings of the Eurographics Workshop on Rendering 2001)*, pages 139–150. Springer Verlag, 2001.

[6] T. Karkanis and A. J. Stewart. Curvature-dependent triangulation of implicit surfaces. *IEEE Computer Graphics and Applications*, 21(2):60–69, 2001.

[7] F. Levet, J. Hadim, P. Reuter, and C. Schlick. Anisotropic sampling for differential point rendering of implicit surfaces. In *WSCG (Winter School of Computer Graphics)*, 2005.

[8] D. Levin. Mesh-independent surface interpolation. In G. Brunnett, B. Hamann, and H. Mueller, editors, *Geometric Modeling for Scientific Visualization*. Springer, Heidelberg, Germany, 2003.

[9] M. D. Meyer, P. Georgel, and R. T. Whitaker. Robust particle systems for curvature dependant sampling of implicit surfaces. In *SMI (Shape Modeling Internatinal)*, 2005.

[10] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization 2002*, pages 163–170, 2002.

[11] A. Rosch, M. Ruhl, and D. Saupe. Interactive visualization of implicit surfaces with singularities. *Eurographics Computer Graphics Forum*, 16(5):295–306, 1997.

[12] W. Y. Su and J. C. Hart. A programmable particle system framework for shape modeling. In *SMI (Shape Modeling Internatinal)*, 2005.

[13] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 26(2):185–194, 1992.

[14] G. Turk. Generating textures for arbitrary surfaces using reaction-diffusion. *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, 25(4):289–298, 1991.

[15] G. Turk. Re-tiling polygonal surfaces. *Computer Graphics*, 26(2):55–64, 1992.

[16] G. Turk and J. O'Brien. Variational implicit surfaces. Technical Report GIT-GVU-99-15, Georgia Institute of Technology, 1998.

[17] A. P. Witkin and P. S. Heckbert. Using particles to sample and control implicit surfaces. In *Proceedings of ACM SIGGRAPH 94*, pages 269–278, 1994.

[18] 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, 1998.