Poisson Sphere Distributions

Ares Lagae & Philip Dutré

Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200 A, B-3001 Heverlee, Belgium
Email: {ares.lagae,philip.dutre}@cs.kuleuven.be

Abstract

We present a method for efficiently generating Poisson sphere distributions, the three-dimensional equivalent of Poisson disk distributions. Our method is able to generate arbitrarily large Poisson sphere distributions in real time, and allows local evaluation of the generated distributions. The method we present is based on cubes with colored corners, called corner cubes, an extension of corner tiles. We generalize tiling algorithms and construction methods from corner tiles to corner cubes. Poisson sphere distributions have several useful applications in computer graphics, such as geometry instancing and a three-dimensional procedural object distribution function.

1 Introduction

Well-distributed point sets have several applications in computer graphics, such as modeling, sampling, halftoning and non-photorealistic rendering. These applications often rely on efficient methods for generating well-distributed point sets.

One of the most popular two-dimensional well-distributed point sets is the Poisson disk distribution. Recently, several techniques have been proposed for the efficient generation and evaluation of Poisson disk distributions. In this paper, we extend this work to three dimensions. We introduce the Poisson sphere distribution, the three-dimensional equivalent of the Poisson disk distribution.

We present a method for efficiently generating Poisson sphere distributions of arbitrary size in real time. The generated distribution can be evaluated locally. We discuss several applications of Poisson sphere distributions, including geometry instancing and a three-dimensional procedural object distribution function, a new texture basis function that distributes procedurally generated objects over a procedurally generated solid texture.

Our method is based on cubes with colored corners, called corner cubes. Corner cubes are a generalization of corner tiles [13], related to Wang tiles and Wang cubes. In this paper we define corner cubes and extend tiling algorithms and construction methods for corner tiles to corner cubes.

2 Background and Related Work

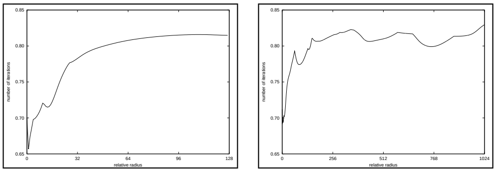
Poisson disk distributions were first used in computer graphics in the context of sampling. Dippé and Wold [5], Cook [3] and Mitchell [16] introduced non-uniform sampling and the Poisson disk distribution to solve the aliasing problem.

Poisson disk distributions are traditionally generated using an expensive dart throwing algorithm [3]. Fast methods that generate approximate Poisson disk distributions were proposed by various authors [5, 17]. McCool and Fiumé [15] introduced relaxation dart throwing and Lloyd's relaxation scheme to generate well distributed point sets.

Dippé and Wold [5] already in 1985 suggested tile-based methods for efficiently generating Poisson disk distributions. Since then, several tile-based methods were proposed. Most of them use Wang tiles [25, 7], square tiles with colored edges. Shade et al. [22] extended the dart throwing method to Wang tiles. Hiller et al. [8] presented a method based on Lloyd's relaxation scheme, which was later adopted by Cohen et al. [2]. Ostromoukhov et al. [19] presented an interesting method based on the Penrose tiles [7] and precomputed relaxation vectors. Lagae and Dutré introduced improved construction methods for Poisson disk tiles based on Wang tiles [12] and corner tiles [13]. Kopf et al. [11] use recursive Wang tiles to generate varying density Poisson disk distributions. Jones [10] and Dunbar and Humphreys [6] presented fast implementations of the dart throwing algorithm.

Next to sampling, Poisson disk distributions have several other applications in computer graphics, such as geometry instancing [4], object distribution for illustration [21], and procedural texturing [12].

Wang tiles are used frequently in computer graph-



(a) (b)

Figure 1: The relative radius versus the number iterations of Lloyd's relaxation scheme applied to (a) a Poisson disk distribution and (b) a Poisson sphere distribution generated with dart throwing. Note the local minima in (b).

ics for generating non-periodic textures [23, 18, 2]. Wang cubes were introduced in the context of discrete mathematics [9], and have recently been used for example-based volume illustrations [14].

3 Poisson Sphere Distributions

A Poisson sphere distribution is a three-dimensional random point distribution in which all points are separated by a minimum distance. Half that distance is called the radius of the distribution. If a sphere of that radius is placed at each point, then no two spheres will intersect.

In this section we discuss radius specification, generation and analysis of Poisson sphere distributions.

3.1 Radius Specification

In [12], we have introduced a convenient radius specification scheme for two-dimensional point distributions. The densest packing of disks in the plane is the hexagonal lattice, with a packing density η of

$$\eta = \frac{\pi}{\sqrt{12}} \approx 0.9069. \tag{1}$$

The packing density is defined as the fraction of the area filled by the disks. When distributing N disks over a unit toroidal domain, the maximum disk area is thus η/N. The maximum disk radius r_{max} is therefore given by

$$r_{max} = \sqrt{\frac{1}{2\sqrt{3}N}}. \tag{2}$$

The Poisson disk radius r can now be specified as a fraction $\rho \in [0 \ldots 1]$ of the maximum disk r_{max}

$$r = \rho r_{max}. \tag{3}$$

The relative radius ρ is independent of the size of the domain and of the number of points. It is a measure of the quality of the Poisson disk distribution. The higher ρ, the closer the densest possible packing is approached. Good distributions should have a relative radius that is high ($\rho > 0.70$), but not too high either ($\rho < 0.85$), because regularity should be avoided.

The relative radius specification scheme generalizes to three dimensions. The packing density η of the densest packing of spheres is given by

$$\eta = \frac{\pi}{\sqrt{18}} \approx 0.7405. \tag{4}$$

In three dimensions, the packing density is defined as the fraction of the volume filled by the spheres. This problem, also known as *the Kepler Problem*, was solved in 1998 [1]. In three dimensions, the maximum sphere radius is therefore

$$r_{max} = \sqrt[3]{\frac{1}{4\sqrt{2}N}}. \tag{5}$$

As in two dimensions, good Poisson sphere distributions should have have a relative radius that is relatively high.

3.2 Generation

Most algorithms for generating Poisson disk distributions generalize to three dimensions. We briefly discuss dart throwing [3], relaxation dart throwing [15], and Lloyd's relaxation scheme [15].

Dart Throwing The dart throwing algorithm generates uniformly distributed points, and rejects points that do not satisfy the minimum separation with already generated points. Practice show that dart throwing can be used for generating Poisson sphere distributions with a relative radius up to 0.70.

Relaxation Dart Throwing The relaxation dart throwing algorithm places points with a large radius initially. If no more space is found for a large number of attempts, the radius is reduced by some fraction before new points are added. This algorithm is faster and easier to control than dart throwing. With an initial radius of 0.15, 10000 attempts and a radius reduction fraction of 0.99, the relative radius of the distributions is about 0.70.

Lloyd's Relaxation Scheme Lloyd's relaxation scheme is an iterative process. In each iteration, the Voronoi diagram of the point set is computed, and all points are moved to the centroid of their Voronoi cell. This scheme can be used to improve the radius of a distribution, or to generate a well-distributed point set from a uniform point distribution. It is the only method that allows to generate distributions with a high relative radius (up to 0.85). The method also works in three dimensions, but does not converge as easily as in two dimensions. This is illustrated in Figure 1. Although the global trend indicates convergence, the method seems to get stuck in local minima often. We believe this is because the maximum radius configurations are more stable in two dimensions than in three. As a consequence, much more iterations are needed in three dimensions.

3.3 Analysis

Radius statistics and spectral analysis are the two primary means to analyze Poisson disk and Poisson sphere distributions. We use a generalization of Bartlett's method of averaging periodograms [24]. The power spectrum of a method for generating Poisson sphere distributions is estimated by averaging Fourier transforms (periodograms) of distributions generated with the method. Similar to Poisson disk distributions, the power spectrum of Poisson sphere distributions should be radially symmetric. Therefore, the power spectrum is reduced to two one-dimensional measures: radially averaged power and anisotropy. The power should exhibit the typical blue noise profile, and the anisotropy should be low, indicating good radial symmetry. Figure 2 shows the typical power spectrum of Poisson sphere distributions.

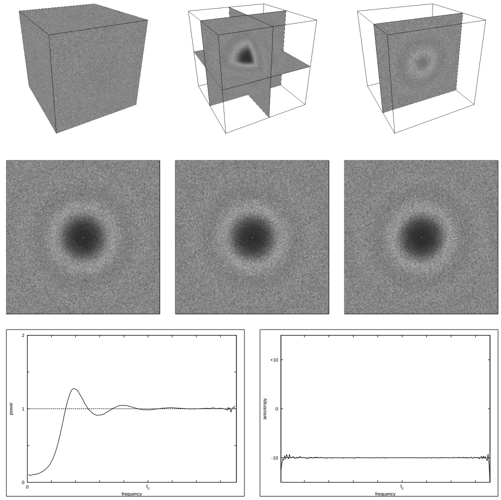


Figure 2: Spectral analysis of Poisson sphere distributions of 65536 points and a relative radius of 0.70 generated with dart throwing. The top row shows several slices of the power spectrum. The middle row shows the three coordinate plane slices. The bottom rows shows the radially averaged power and anisotropy.

4 Tile Based Generation

In this section we present our method for efficiently generating arbitrary large Poisson sphere distributions. We first introduce corner cubes and present a tiling algorithm for corner cubes. We then show how to construct a Poisson sphere distribution over a set of corner cubes. We end this section with an analysis of the tiled Poisson sphere distributions generated with our method.

4.1 Corner Cubes

Corner cubes are an extension of corner tiles [13] to three dimensions. Corner cubes are cubic tiles with colored corners. Similar to Wang tiles, the cubes have a fixed orientation. Because a cube has 8 corners, a complete set of corner cubes over C colors consists of C^8 cubes. Figure 3 shows some of the cubes of the complete tile set over two colors, counting 256 cubes.

A tiling is generated by placing the cubes next to each other such that adjoining corners have matching colors. Each cube in the set can be used arbitrarily many times. Figure 4 shows an example tiling generated with the corner cube set over two colors. The cubes are enumerated using their corner colors. The eight corner colors are interpreted as an 8-digit base-C number. A base conversion converts the corner colors to a cube index and vice versa. Figures 3 and 4 also show the cube indices.

4.2 Tiling Algorithm

The direct stochastic tiling algorithm of [13] extends nicely to three dimensions. Assuming that the cubes are placed with their corners on the integer lattice points, a tiling of space is equivalent with a coloring of the integer lattice. A stochastic tiling is thus obtained by assigning a random color to each integer lattice point. This can easily be accomplished with a hash function. We have obtained good results using a hash function based on a permutation table [20].

This tiling algorithm is very efficient. Only eight evaluations of the hash function and a base conversion are needed to decide which cube is at a given location. Moreover, the algorithm allows local evaluation of the tiling: it is able to decide which cube is at a given location without computing and storing neighboring cubes. Consistency with neighboring cubes is guaranteed by the hash function.

4.3 Cube Construction

For constructing a Poisson disk distribution over a set of cubes, we follow the strategy of [12] and [13]. We use corner cubes because Wang cubes are subject to the corner problem [13], and because extending the approach of [12] would result in a set of millions of cubes.

The Poisson disk radius determines different regions in a tile: corner regions, edge regions, face regions and an interior region. These regions are shown in Figure 5. Points in these regions affect points in respectively seven, three, one and zero neighboring tiles. The tile regions are modified such that the distance between regions of the same type is at least twice the Poisson sphere radius. The modified tile regions are shown in Figure 6. This allows us to treat different regions of the same type independent. The modified tile regions give rise to a dual tiling. This is illustrated in Figure 7. The four kinds of tiles in the dual tiling are corner tiles, edge tiles, face tiles and interior tiles. Figure 8 shows the shape of these tiles. These four kinds of tiles correspond respectively to the combination of eight modified corner regions, four modified edge regions, two modified face regions, and an interior region. Note that the corner tile is a great rhombicuboctahedron, an Archimedean solid.

Constructing a Poisson sphere distribution over a set of corner cubes is done using the dual tiles. This process is illustrated in Figures 9, 10, 11 and 12. First, a Poisson sphere distribution is constructed over the corner tiles (Figure 9(a)). For each corner tile, a Poisson sphere distribution is generated over the toroidal unit cube using relaxation dart throwing (Figure 9(b)) optionally followed by Lloyd's relaxation scheme (Figure 9(c)). All spheres with their center inside the corner tile are then cut out of the distribution (Figure 9(d)). Next, a Poisson sphere distribution is constructed over the edges tiles by assembling the corresponding corner tiles (Figure 10(a)), generating a Poisson sphere distribution using relaxation dart throwing (Figure 10(b)) optionally followed by Lloyd's relaxation scheme (Figure 10(c)), and cutting out the edge tile (Figure 10(d)). No new points are added to the corner tiles, and the points of the corner tiles remain fixed during relaxation. Then, a Poisson sphere distribution is constructed over the face tiles in the same way (Figure 11). Finally, the corner cubes are constructed. This is done by assembling all corresponding corner tiles, edge tiles and face tiles (Figure 12(a)). All points further away from the cube surface than the Poisson sphere radius are discarded (Figure 12(b)). A Poisson sphere distribution is generated using relaxation dart throwing, both in the inside (Figure 12(c)) and outside (Figure 12(d)) of the cube, optionally followed by Lloyd's relaxation scheme (Figure 12(e)). The corner cube is then cut out of the point distribution (Figure 12(f)).

In practice, this construction method works very well. Our method is capable of constructing sets of cubes with a relative radius up to 0.80. We have also verified that the point density in the different regions corresponds to the expected number of points in the regions according to their volume. Note that the construction of the cubes is a preprocess. Constructing a set of cubes takes about an hour on a regular computer. The most expensive operation is Lloyd's relaxation. One set of cubes is sufficient to generate as much Poisson disk distributions as required, due to the non-periodic tiling algorithm.

4.4 Analysis

Figure 13 shows the spectral analysis of tiled Poisson sphere distributions. The power spectrum is very similar to the power spectrum of non-tiled Poisson sphere distributions. The underlying tiling is revealed through a grid of spikes. This is typical for tiled point distribution. The radially averaged power exhibits the typical blue noise spectrum. Compared with the power spectrum of non-tiled Poisson disk distributions, the anisotropy is too high. However, the approach we present in this paper is currently the only tile-based approach that generalizes to three dimensions.

5 Applications

Poisson sphere distributions have various applications in computer graphics. In this section we discuss procedural modeling, procedural texturing, and suggest several other applications. For most of these applications, Poisson sphere distributions are the only option, because other well distributed point sets, such as low-discrepancy point sets, have a too small relative radius and are too regular.

Geometry Instancing Many natural and man-made distributions follow a Poisson sphere like pattern that can be modeled using corner cubes. Because the underlying tiling can be evaluated locally,

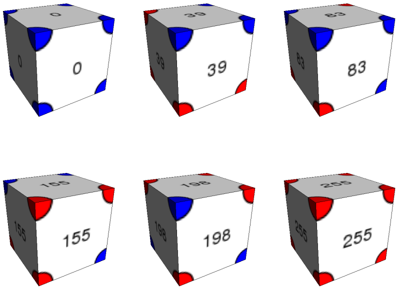


Figure 3: A number of cubes from the complete set of corner cubes over two colors.

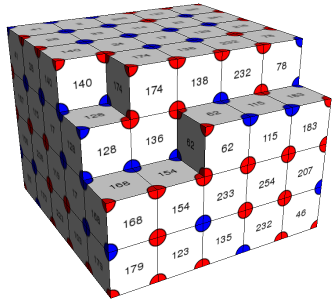


Figure 4: A tiling generated from the complete set of corner cubes over two colors.

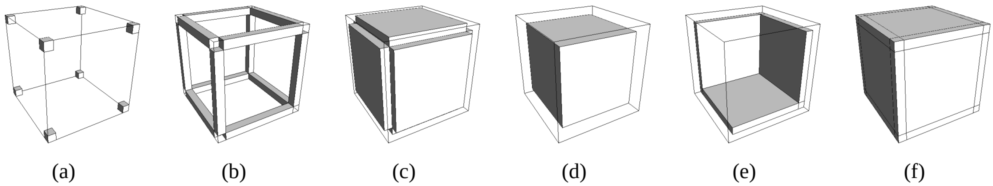


Figure 5: The Poisson sphere radius determines (a) corner regions, (b) edge regions, (c) face regions, and (d) an interior region. Also shown is (e) an inside view and (f) an outside view of the cube.

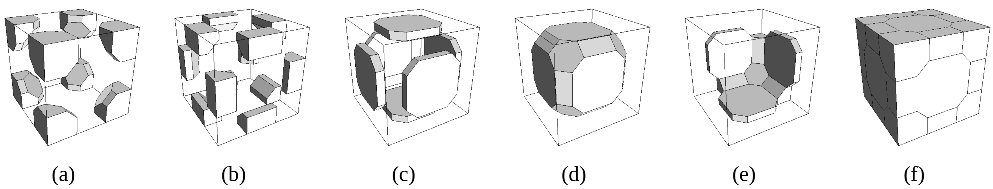


Figure 6: The tile regions are modified to facilitate the construction of a Poisson disk distribution over a set of tiles. The (a) modified corner regions, (b) modified edge regions, (c) modified face regions, and (d) modified interior region are shown. Also shown is (e) an inside view and (f) an outside view of the cube.

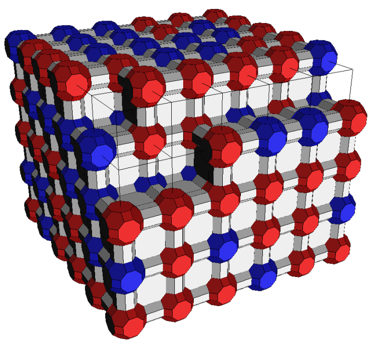


Figure 7: The dual tiling implied by the modified tile regions (generated from the tiling shown in Figure 4).

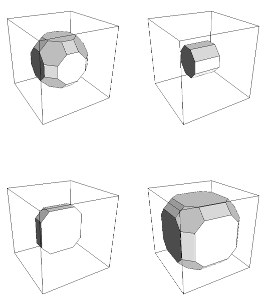
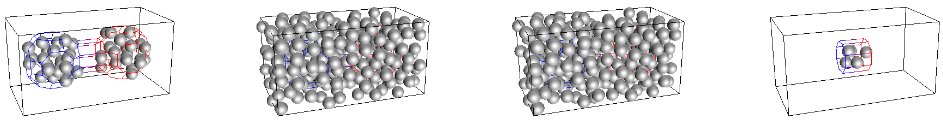


Figure 8: The four kinds of tiles in the dual tiling are corner tiles, edge tiles, face tiles and interior tiles.

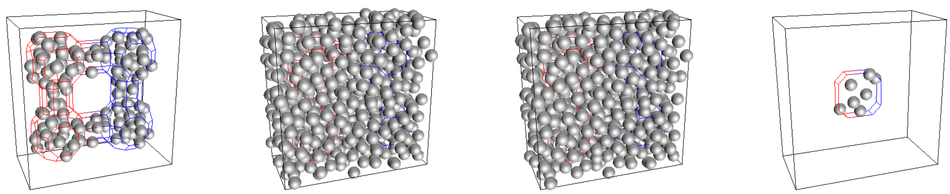
(a) (b) (c) (d)

Figure 9: Construction of a Poisson sphere distribution over a corner tile.



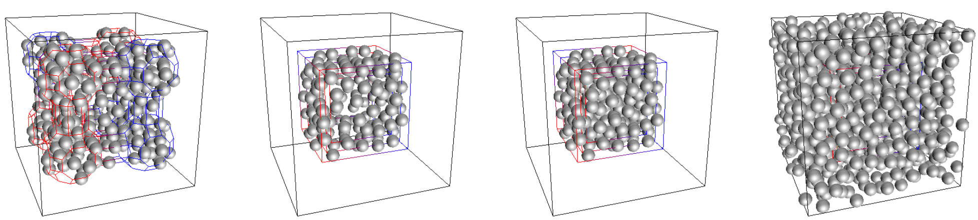
(a) (b) (c) (d)

Figure 10: Construction of a Poisson sphere distribution over an edge tile.

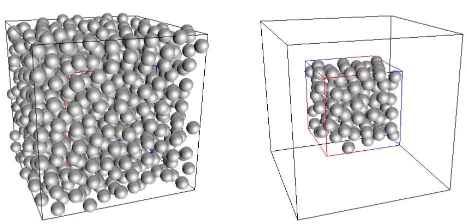


(a) (b) (c) (d)

Figure 11: Construction of a Poisson sphere distribution over a face tile.



(a) (b) (c) (d)



(e) (f)

Figure 12: Construction of a Poisson sphere distribution over a corner cube.

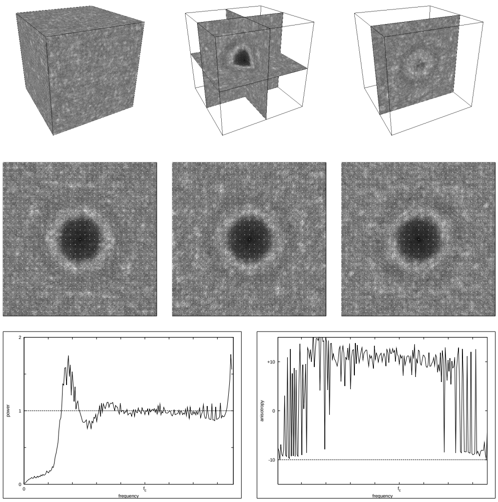


Figure 13: Spectral analysis of Poisson sphere distributions of 65536 points and a relative radius of 0.75 generated by tiling $8 \times 8 \times 8$ tiles of a tile set with 128 points per tile. The top row shows several slices of the power spectrum. The middle row shows the three coordinate plane slices. The bottom rows shows the radially averaged power and anisotropy.

our method allows to instantiate geometry on-the-fly. Rather than storing instancing information, instances are generated procedurally. This is a significant advantage for geometry instancing. Figure 14 shows Saturn's asteroid belt, modeled by cutting out a ring of points from a Poisson sphere distribution. When flying through the asteroid belt, only asteroids within the view frustrum need to be instantiated.

Procedural Texturing The two-dimensional procedural object distribution function [12] can be extended to a solid version using corner cubes and Poisson sphere distributions. The texture basis function distributes procedurally generated objects over a procedurally generated background. The objects do not to overlap, and the scale, size and orientation of the objects can be easily manipulated. Figure 15 shows the outputs of the three-dimensional texture basis function. Figure 16 shows several textures generated with the distribution. The accompanying video shows how we integrated the texture basis function into a commercial rendering system. The new texture basis function is good at modeling natural materials with particle distributions, such as

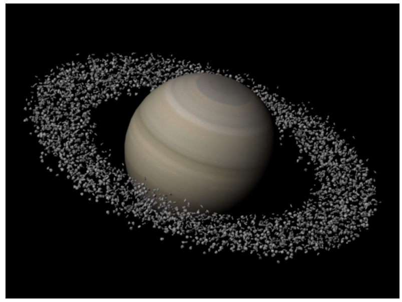


Figure 14: Saturn's asteroid belt was modeled by instantiating several thousands asteroids using a Poisson sphere distribution.

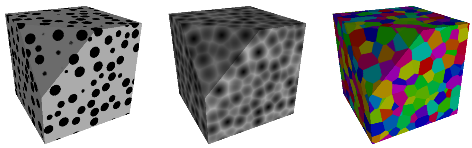
granite, and abstract man-made patterns. The texture basis function has a small memory footprint and is quite efficient: one evaluation is as expensive as 20 evaluations of Perlin's Noise function [20].

Other Applications We believe that Poisson sphere distributions are useful for several other applications, such as rain and snow modeling, volume sampling for representing and rendering volume data, visualizing three-dimensional flow fields, and generating particle distributions for physically based modeling.

6 Conclusion and Future Work

In this paper we have presented an efficient method for generating Poisson sphere distributions, the three-dimensional equivalent of Poisson disk distributions. We have introduced corner cubes, and we have generalized tiling algorithms and construction methods for corner tiles to three dimensions. Our method is capable of producing Poisson sphere distributions of arbitrary size in real time. The distributions can be evaluated locally. We have introduced tools to analyze Poisson sphere distributions. We have discussed several applications of Poisson sphere distributions, including geometry instancing and a three-dimensional object distribution function, a new solid texture basis function.

There are several opportunities for future work. The most challenging one is to develop methods to generate well-distributed point sets in real time according to a given density. It would also be interesting to explore three-dimensional variants of other applications of Wang tiles, such as texture synthesis. Finally, we are interested in working out the other applications mentioned in Section 5.



(a) (b) (c)

Figure 15: The procedural object distribution function returns (a) a boolean indicating whether the point of evaluation is within the Poisson sphere of the closest feature point, (not shown) the coordinates of the feature point, (b) the distance to the feature point, and (c) an unique ID identifying the feature point.

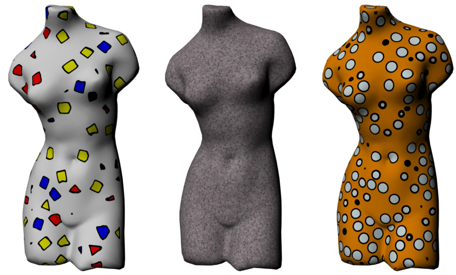


Figure 16: The *Venus* model, carved from solid textures generated with the procedural object distribution function.

Acknowledgments The first author is funded as Aspirant van het Fonds Wetenschappelijk Onderzoek - Vlaanderen. We would like to thank the anonymous reviewers for their constructive remarks. The map of Saturn was created by Björn Jónsson. The 3D asteroid models are courtesy of Scott Hudson, Washington State University.

References

[1] B. Cipra. Packing challenge mastered at last. *Science*, 281, 1998.

[2] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang tiles for image and texture generation. *ACM Trans. Graphics*, 2003.

[3] R. L. Cook. Stochastic sampling in computer graphics. *Computer Graphics (Proc. ACM SIGGRAPH 86)*, 5(1), 1986.

[4] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. *Proc. ACM SIGGRAPH 1998*, 1998.

[5] M. A. Z. Dippé and E. H. Wold. Antialiasing through stochastic sampling. *Computer Graphics (Proc. ACM SIGGRAPH 85)*, 19(3), 1985.

[6] D. Dunbar and G. Humphreys. A spatial data structure for fast poisson-disk sample generation. *ACM Trans. Graphics*, 25(3):503–508, 2006.

[7] B. Grünbaum and G. C. Shepard. *Tilings and Patterns*. W. H. Freeman and Company, 1986.

[8] S. Hiller, O. Deussen, and A. Keller. Tiled blue noise samples. *Proc. Vision Modeling Visualization 2001*, 2001.

[9] K. C. II. and J. Kari. An aperiodic set of wang cubes. *J. Universal Computer Science*, 1(10), 1995.

[10] T. R. Jones. Efficient generation of Poisson-disk sampling patterns. *J. Graphics Tools*, 11(2):27–36, 2006.

[11] J. Kopf, D. Cohen-Or, O. Deussen, and D. Lischinski. Recursive wang tiles for real-time blue noise. *ACM Trans. Graphics*, 25(3):509–518, 2006.

[12] A. Lagae and P. Dutré. A procedural object distribution function. *ACM Trans. Graphics*, 24(4), 2005.

[13] A. Lagae and P. Dutré. An alternative for Wang tiles: Colored edges versus colored corners. *ACM Trans. Graphics*, 2006. Accepted.

[14] A. Lu and D. S. Ebert. Example-based volume illustrations. *IEEE Visualization 2005 - (VIS'05)*, page 83, 2005.

[15] M. McCool and E. Fiume. Hierarchical Poisson disk sampling distributions. *Graphics Interface '92*, 1992.

[16] D. P. Mitchell. Generating antialiased images at low sampling densities. *Computer Graphics (Proc. ACM SIGGRAPH 87)*, 21(4), 1987.

[17] D. P. Mitchell. Spectrally optimal sampling for distribution ray tracing. *Computer Graphics (Proc. ACM SIGGRAPH 91)*, 25(4), 1991.

[18] F. Neyret and M.-P. Cani. Pattern-based texturing revisited. *Proc. ACM SIGGRAPH 1999*, 1999.

[19] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graphics*, 23(3), 2004.

[20] K. Perlin. Improving noise. *ACM Trans. Graphics*, 2002.

[21] A. Secord, W. Heidrich, and L. Streit. Fast primitive distribution for illustration. *EGRW '02: Proc. of the 13th Eurographics workshop on Rendering*, 2002.

[22] J. Shade, M. F. Cohen, and D. P. Mitchell. Tiling layered depth images. Technical report, University of Washington, Department of Computer Science and Engineering, 2000.

[23] J. Stam. Aperiodic texture mapping. Technical Report ERCIM-01/97-R046, European Research Consortium for Informatics and Mathematics (ECRIM), 1997.

[24] R. Ulichney. *Digital halftoning*. MIT Press, Cambridge, MA, USA, 1987.

[25] H. Wang. Proving theorems by pattern recognition II. *Bell Systems Technical J.*, 40, 1961.