EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Sampling with AA Patterns

Abdalla G. M. Ahmed

University of Konstanz

Abstract

We describe a technique for compressing noise sets. We use so-called AA Patterns: ornamental algorithmic-art point sets that are structured in a complex intertwined manner. We optimize these patterns to match a noise profile, and exploit their powerful inherent neighborhood identification mechanism to efficiently store the result of optimization.

1. Introduction

Point sets are ubiquitous in computer graphics. Different applications might have different requirements, but irregular isotropic point sets are universal and suit many applications including sampling, stippling and halftoning, meshing and remeshing, and distributing objects. Such point sets are typically extracted from white noise via expensive refinement processes. The irregularity of these colored noise sets — most commonly blue [Uli88] — is challenging, and many algorithms were suggested trying to reduce the cost of production and/or improve the quality [Coo86, MF92, Jon06, DH06, Wei08, BSD09, GM09, SGBW10, LWSF10, EDP*11, Fat11, SHD11, XLGG11, EMP*12, dGBOD12, ZHWW12, OG12, HSD13, GYJZ15, JZW*15].

The high production cost of noise sets lead to the idea of producing them offline and storing them for later use. This idea of pre-computing is not a new one [DW85], and it also received a lot of attention during the past three decades. The main challenge is that storing such sets is expensive: they are difficult to compress because of their high entropy. But storage is not the only cost involved in handling such noise sets: they are also difficult to retrieve efficiently, since irregularity makes points difficult to index.

The traditional approach for exporting irregular point sets is to lay them in tiles, and assemble these tiles into a tiling at run-time. Tile-based solutions offer a wide range of options that trade quality for price (memory and/or performance), from a single toroidal square tile [Gla95], the cheapest (in both senses of the word), to very sophisticated, carefully designed tile sets such as polyhexes [WPC*14], which offer very good quality but are also very expensive in terms of memory footprint. There are more options in between [CSHD03, ODJ04, KCODL06, LD06, Ost07], but the ratio of quality to table sizes remains low for all tiling solutions we are aware of, not to mention the excessive coding complexity to handle the indexing and retrieval of individual samples, as well as the construction of the tiling.

1.1. Our Idea

We propose a novel solution to the problem of compressing noise sets: instead of trying to compress an arbitrary noise set, we start from a compressible point set and morph it into the desirable noise profile: blue, green, etc. [ZHWW12]. Rather than storing the points themselves we only have to store the required displacements for such morphing.

Algorithmic Art (AA) is probably the right context to search for such compressible point sets, where "algorithmic" implies compressibility of the underlying set, while "art" suplements the high entropy needed for good quality noise. Since our goal is to lay individual samples, we are interested in ornamental AA built from discrete repeating elements that are shuffled in subtle complex manners. Indeed, having repeating elements is essential for a look-up solution, but such repetition is also required to be complex.

We found an excellent candidate in so-called AA Patterns [Ahm11c]. These patterns are not only ornamental but also point-based, which brings us one step closer to our goal. In the rest of this article we describe AA Patterns and explain how they might be morphed into different noise profiles, with very little memory requirements to store morphing information.

2. AA Patterns

"AA Patterns" refers to a parametric point-based algorithmic art that emerges from quantization aliasing artifacts under certain linear transformations [Ahm11c, Ahm11a, Ahm11b, Ahm12]. Specifically, if an image is scanned pixel by pixel, and each pixel (x,y) is projected into a target pixel (X,Y) in a view-port using:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \left\lfloor \frac{1}{2} \begin{bmatrix} \alpha & -1 \\ 1 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right\rfloor \quad \alpha \in (1,2)\,, \tag{1}$$

then some source pixels overlap on the same target, while some target pixels are left empty. $\mathrm{AA}(\alpha)$ is the set of these empty pixels. Following a sequence of transformations, the pattern can be associated directly with the source points using this simple algebraic

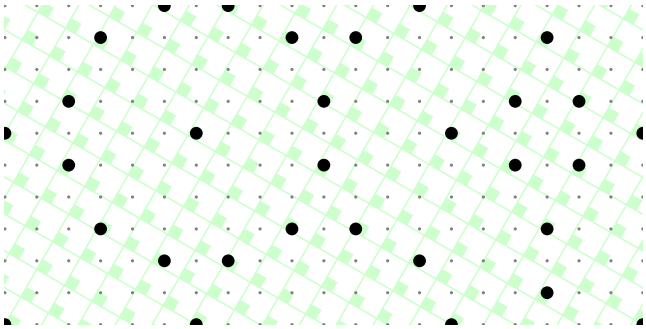

Figure 1: *A graphical representation of Eqs. (2, 3, 4), an appropriately transformed array of unit cells is used to filter points in the integer grid. The index (dX, dY) is the location of a point in the hosting cell, and the size of the set (green) region in each cell is $t \times t$.*

formulation:

$$\mathrm{AA}(\alpha) = \{(x,y) : x,y \in \mathbb{Z}; dX < t; dY < t\}, \tag{2}$$

where

$$\begin{aligned} dX &= \mathrm{dX}(x,y) \triangleq \{(\alpha x + y)/2\} \\ dY &= \mathrm{dY}(x,y) \triangleq \{(\alpha y + x)/2\} \end{aligned}, \tag{3}$$

and

$$t = (\alpha - 1)/2. \tag{4}$$

Thus, an AA Pattern is a subset of a regular grid: each grid point (x,y) is mapped to an index (dX, dY) in the unit torus using Eq. (3), then points are filtered by testing their indexes against a certain threshold defined in Eq. (4). We call the $t \times t$ block of valid indexes the "set region"; see Figure 1.

3. Neighborhood Indexing in AA Patterns

An important property of the index mapping (Eq. (3)) is that it is linear; that is, any pattern-space translation corresponds to a specific index-space translation[†]:

$$\mathrm{dX}(x + \Delta_x, y + \Delta_y) = \mathrm{dX}(x,y) + \mathrm{dX}(\Delta_x, \Delta_y).$$

An important consequence is that any arrangement of points in the grid (a triangle, square, cross, whatever) has an index-space image: a specific arrangement of indexes, and that image is unique for each arrangement, up to wrapping around the edge of the unit torus. Now an arrangement of points exists in the pattern iff its whole image fits in the set region; see Figure 2.

From here we may see that any arrangement of points found in the pattern repeats infinitely many times. There are two possibilities for this: when α is a rational number p/q, the index-space is a finite $2q \times 2q$ set, and the pattern is periodic; hence each and every detail repeats periodically. On the other hand, when α is irrational indexes are unique, and the pattern is aperiodic. Even though indexes are dense, the index (t,t) never exists, therefore any image

† Mind that all index-space arithmetic is modulo 1.

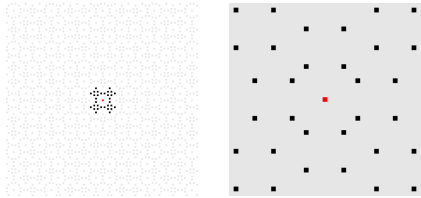

Figure 2: *An arrangement of grid points (left, black and red) and its index-space image (right). The image completely fits inside the set region (gray), indicating that the corresponding arrangement of points is a valid subset of the pattern.*

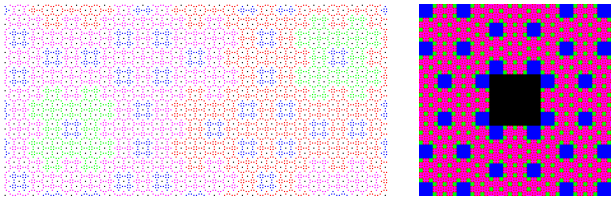

Figure 3: *Painting similar "clusters" of points in similar colors corresponds to complex fractal index-space maps. The density of each cluster in the pattern is determined by the margin that its image leaves in the set region. An algorithm for retrieving such maps is described in [Ahm11c].*

that fits in the set region leaves some room for translation. Thus, any index-space image repeats infinitely many times, and the corresponding arrangement of points repeats infinitely many times in the pattern. Note that small index-space translations correspond to large pattern-space translations.

Index-space images of individual pattern-space arrangements turn into index-space maps when all similar arrangements are addressed. For example, when all distinguishable "clusters" of similar shapes are painted similarly, it reflects as a fractal index-space map; see Figure 3.

Having repeating elements is important for look-up methods, and being able to identify similar neighborhoods is the key for efficiently tabulating optimization results. For example, the overall displacement of a point in n iterations of Lloyd's relaxation [MF92, Llo82] is — roughly speaking — determined by the initial relative positions of n rings of neighbors around that point. We are therefore interested in identifying identical square neighborhoods of a certain size around points in the pattern.

Unlike the fractal coloring maps (Figure 3), rectangular neighborhood around points are identified by simple (checkered) index-space maps like the one in Figure 4. Retrieving such maps is described in detail in [AHD15]. Once such a map is available, it can be used to record the displacement of points during optimization, since points with identical neighborhoods tend to respond similarly to common optimization algorithms. For vanilla Lloyd's relaxation the behavior of points automatically follows the map, but for other optimizations we might need to use the map to influence the process, much like the use of structural indexes in tile-based methods [ODJ04, Ost07, WPC*14].

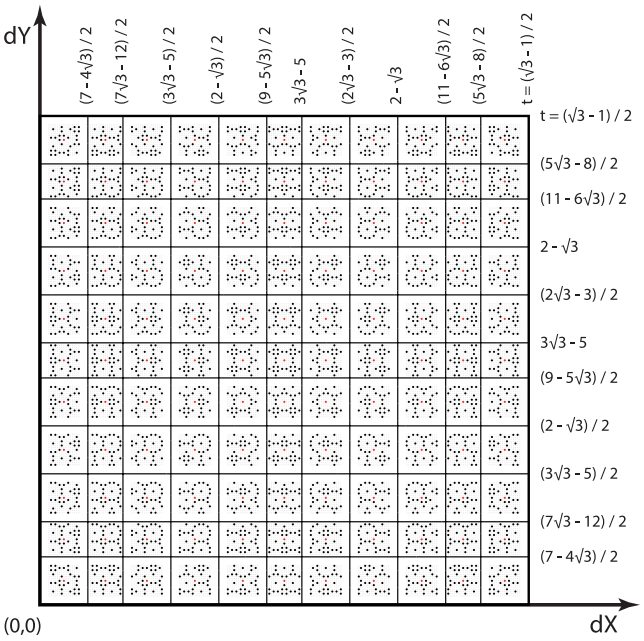

Figure 4: *Ranges of indexes in* AA $(\sqrt{3})$*, and the corresponding neighborhood around indexed points. The map is multi-resolution: each block is subdivided into smaller blocks if a wider neighborhood is considered. Indeed, the whole pattern around a point is known if the exact index is given.*

Actual optimization is performed only on a small portion of the pattern, since it is provable that all distinct neighborhoods in the pattern exist in only a small part of it. Even better, it is provable that there exists a small periodic AA Pattern with the same set of distinct neighborhoods as any infinite aperiodic AA Pattern.

4. Summary of Results

Neighborhood maps are inherent in AA Patterns, they do not have to be tabulated. These maps are hierarchical: they can be retrieved to any desirable resolution, corresponding to arbitrary large neighborhoods around the points. The size of the map grows linearly with the size of identified neighborhood, in sharp contrast to tile-based methods where the number of distinct neighborhoods grows exponentially. We are unaware of a tiling method that can index even the second ring of neighbors and stay within practical limits of resources. In AA Patterns we may index more than 20 rings of neighbors using few kilobytes maps. This leads to much better quality combined with much less resources.

With a production rate of 100M points per second in commodity hardware, our method is an order of magnitude faster than the fastest known tiling method. Another subtle advantage is the full random access to points. This is important for some applications like distributing objects.

5. Future Perspectives

We plan to extend the concept to adaptive generation of samples. AA Patterns are self-similar by nature [Ahm12], but so far we faced two obstacles: that this self-similarity is topological but not geometrical, and that it is met upon zooming-out, not zooming-in. The two problems seem to be inter-related.

Currently AA Patterns are defined only in two dimensions. We are investigating ways to extend them — hence the sampling concept based on them — to higher dimensions.

References

[AHD15] AHMED A. G. M., HUANG H., DEUSSEN O.: Aa patterns for point sets with controlled spectral properties. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 212:1–212:8. 2

[Ahm11a] AHMED A. G. M.: AA Patterns. In *EG 2011 - Posters* (Llandudno, UK, April 2011), Eurographics Association, pp. 35–36. Accessible online at http://diglib.eg.org/EG/DL/conf/EG2011/posters/035-036.pdf. 1

[Ahm11b] AHMED A. G. M.: Mathematical Hints for Parameter Selection for AA Patterns. In *Bridges Coimbra* (Coimbra, Portugal, July 2011), The Bridges Organization, pp. 271–278. 1

[Ahm11c] AHMED A. G. M.: Pixel Patterns from Quantization Artifacts of Forward Affine Mapping. *Journal of Graphics, GPU, and Game Tools 15*, 2 (2011), 73–94. 1, 2

[Ahm12] AHMED A. G. M.: On the Fractal Behaviour of AA Patterns. In *Theory and Practice of Computer Graphics* (Rutherford, United Kingdom, September 2012), Hamish Carr and Silvester Czanner, (Ed.), Eurographics Association, pp. 93–97. 1, 3

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of lloyd's method. In *ACM SIGGRAPH 2009 Papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 86:1–86:8. 1

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1 (Jan. 1986), 51–72. 1

[CSHD03] COHEN M. F., SHADE J., HILLER S., DEUSSEN O.: Wang tiles for image and texture generation. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 287–294. 1

[dGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 171:1–171:11. 1

[DH06] DUNBAR D., HUMPHREYS G.: A spatial data structure for fast poisson-disk sample generation. *ACM Trans. Graph. 25*, 3 (July 2006), 503–508. 1

[DW85] DIPPÉ M. A. Z., WOLD E. H.: Antialiasing through stochastic sampling. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1985), SIGGRAPH '85, ACM, pp. 69–78. 1

[EDP*11] EBEIDA M. S., DAVIDSON A. A., PATNEY A., KNUPP P. M., MITCHELL S. A., OWENS J. D.: Efficient maximal poisson-disk sampling. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 49:1–49:12. 1

[EMP*12] EBEIDA M. S., MITCHELL S. A., PATNEY A., DAVIDSON A. A., OWENS J. D.: A simple algorithm for maximal poisson-disk sampling in high dimensions. *Comp. Graph. Forum 31*, 2pt4 (May 2012), 785–794. 1

[Fat11] FATTAL R.: Blue-noise point sampling using kernel density model. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 48:1–48:12. 1

[Gla95] GLASSNER A. S.: *Principles of digital image synthesis: Vol. 1*, vol. 1. Elsevier, 1995. 1

[GM09] GAMITO M. N., MADDOCK S. C.: Accurate multidimensional poisson-disk sampling. *ACM Trans. Graph. 29*, 1 (Dec. 2009), 8:1–8:19. 1

[GYJZ15] GUO J., YAN D.-M., JIA X., ZHANG X.: Efficient maximal poisson-disk sampling and remeshing on surfaces. *Computers & Graphics 46* (2015), 72–79. 1

[HSD13] HECK D., SCHLÖMER T., DEUSSEN O.: Blue noise sampling with controlled aliasing. *ACM Trans. Graph. 32*, 3 (July 2013), 25:1–25:12. 1

[Jon06] JONES T. R.: Efficient generation of poisson-disk sampling patterns. *Journal of graphics, gpu, and game tools 11*, 2 (2006), 27–36. 1

[JZW*15] JIANG M., ZHOU Y., WANG R., SOUTHERN R., ZHANG J. J.: Blue noise sampling using an sph-based method. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 211:1–211:11. 1

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive wang tiles for real-time blue noise. *ACM Trans. Graph. 25*, 3 (July 2006), 509–518. 1

[LD06] LAGAE A., DUTRÉ P.: An alternative for wang tiles: Colored edges versus colored corners. *ACM Trans. Graph. 25*, 4 (Oct. 2006), 1442–1459. 1

[Llo82] LLOYD S. P.: Least squares quantization in pcm. *Information Theory, IEEE Transactions on 28*, 2 (1982), 129–137. 2

[LWSF10] LI H., WEI L.-Y., SANDER P. V., FU C.-W.: Anisotropic blue noise sampling. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 167:1–167:12. 1

[MF92] MCCOOL M., FIUME E.: Hierarchical poisson disk sampling distributions. In *Proceedings of the Conference on Graphics Interface '92* (San Francisco, CA, USA, 1992), Morgan Kaufmann Publishers Inc., pp. 94–105. 1, 2

[ODJ04] OSTROMOUKHOV V., DONOHUE C., JODOIN P.-M.: Fast hierarchical importance sampling with blue noise properties. In *ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), SIGGRAPH '04, ACM, pp. 488–495. 1, 2

[OG12] ÖZTIRELI A. C., GROSS M.: Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 170:1–170:10. 1

[Ost07] OSTROMOUKHOV V.: Sampling with polyominoes. *ACM Trans. Graph. 26*, 3 (July 2007). 1, 2

[SGBW10] SCHMALTZ C., GWOSDEK P., BRUHN A., WEICKERT J.: Electrostatic halftoning. *Computer Graphics Forum 29*, 8 (2010), 2313–2327. 1

[SHD11] SCHLÖMER T., HECK D., DEUSSEN O.: Farthest-point optimized point sets with maximized minimum distance. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics* (New York, NY, USA, 2011), HPG '11, ACM, pp. 135–142. 1

[Uli88] ULICHNEY R.: Dithering with blue noise. *Proceedings of the IEEE 76*, 1 (Jan 1988), 56–79. 1

[Wei08] WEI L.-Y.: Parallel poisson disk sampling. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 20:1–20:9. 1

[WPC*14] WACHTEL F., PILLEBOUE A., COEURJOLLY D., BREEDEN K., SINGH G., CATHELIN G., DE GOES F., DESBRUN M., OSTROMOUKHOV V.: Fast tile-based adaptive sampling with user-specified fourier spectra. *ACM Trans. Graph. 33*, 4 (July 2014), 56:1–56:11. 1, 2

[XLGG11] XU Y., LIU L., GOTSMAN C., GORTLER S. J.: Capacity-constrained delaunay triangulation for point distributions. *Computers & Graphics 35*, 3 (2011), 510 – 516. Shape Modeling International (SMI) Conference 2011. 1

[ZHWW12] ZHOU Y., HUANG H., WEI L.-Y., WANG R.: Point sampling with general noise spectrum. *ACM Trans. Graph. 31*, 4 (July 2012), 76:1–76:11. 1