Sampling with Polyominoes

Victor Ostromoukhov
Université de Montréal

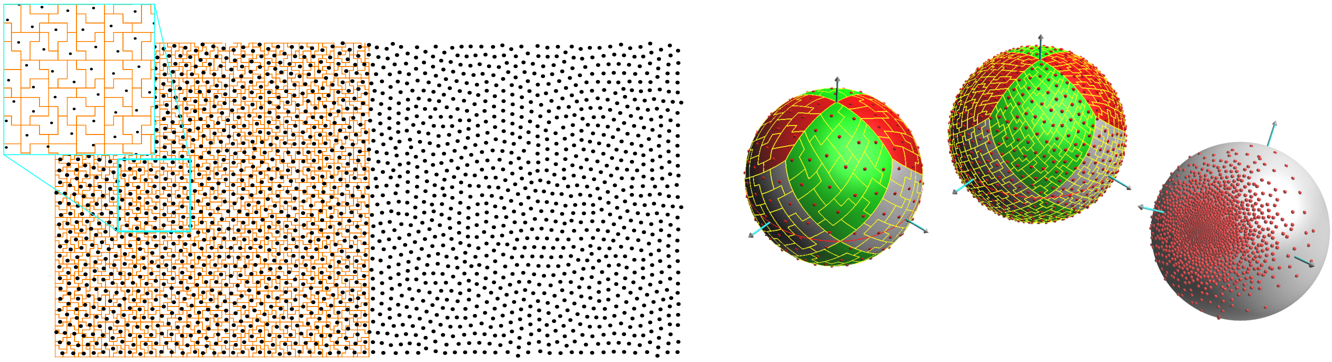


Figure 1: *(Left) Each sampling point of this blue-noise distribution sits on exactly one polyomino. (Right) A set of equiareal polyominoes can be directly mapped on a sphere and hierarchically subdivided (two consecutive levels of subdivision are shown). The method allows fast low-noise low-artifact importance sampling of arbitrary HDR functions (a Gaussian spot is shown on the rightmost sphere).*

Abstract

We present a new general-purpose method for fast hierarchical importance sampling with blue-noise properties. Our approach is based on self-similar tiling of the plane or the surface of a sphere with rectifiable polyominoes. Sampling points are associated with polyominoes, one point per polyomino. Each polyomino is recursively subdivided until the desired local density of samples is reached. A numerical code generated during the subdivision process is used for thresholding to accept or reject the sample. The exact position of the sampling point within the polyomino is determined according to a structural index, which indicates the polyomino's local neighborhood. The variety of structural indices and associated sampling point positions are computed during the offline optimization process, and tabulated. Consequently, the sampling itself is extremely fast. The method allows both deterministic and pseudo-non-deterministic sampling. It can be successfully applied in a large variety of graphical applications, where fast sampling with good spectral and visual properties is required. The prime application is rendering.

Keywords: Importance sampling, Blue noise, Polyominoes, Nonperiodic tiling, Deterministic sampling.

1 Rationale

Sampling is universally used in computer graphics. Hundreds of articles are devoted to studying important properties and limitations of sampling. It is generally accepted today that sampling with blue-noise properties is preferable, for many reasons: for avoiding aliasing, for producing visually satisfactory artifact-free distributions, etc. For a thorough discussion about the role of blue-noise distributions in computer graphics, we refer the interested reader to a recent comprehensive compendium [Pharr and Humphreys 2004].

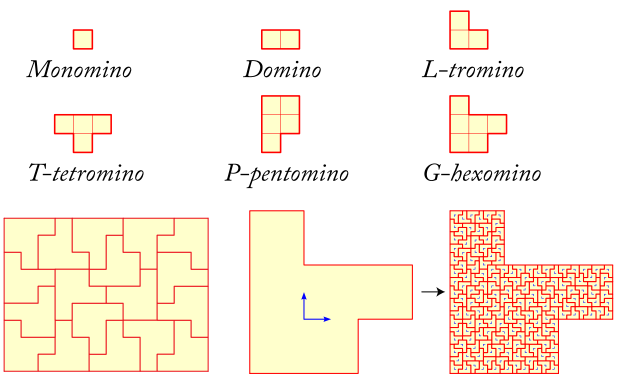


Figure 2: *(Top) Some polyomino shapes. (Bottom-left) A* 12×9 *rectangle filled with 18 G-hexominoes. (Bottom-right) Production rule used for generating tiling with* 9^2*-rep G-hexominoes.*

Very recently, a family of very fast techniques for the generation of blue-noise or Poisson-disc-like distributions have appeared [Ostromoukhov et al. 2004; Kopf et al. 2006; Dunbar and Humphreys 2006; Lagae and Dutré 2006]. All of them work in almost-linear time, with respect to the number of samples, with very low computational cost per sample. Each of the cited techniques has important advantages and limitations. Namely, Penrose tiling-based sampling [Ostromoukhov et al. 2004] shows rather strong artifacts (residues of the tiling's own frequencies) in Fourier spectra, which may become harmful in rendering. Boundary sampling [Dunbar and Humphreys 2006] and corner Wang tiling-based sampling [Lagae and Dutré 2006] do not offer any mechanism for smooth variation of the sample density as a function of arbitrary importance. Finally, recursive Wang tiling-based sampling [Kopf et al. 2006] produces a higher level of noise, compared to the other techniques (see Figure 7). Another important limitation of the latter technique consists in the large number of samples per tile (thousands, as presented in their paper). This is obviously an obstacle for hierarchical rendering algorithms, where the total number of samples is only a few dozens, and where the treatment of each sample, even rejected, has a cost.

The method presented in this paper helps to overcome the limitations of all cited approaches. We build our approach on Penrose-based sampling, with two notable differences: we use rectifiable polyominoes instead of Penrose tiling, and we use conventional base-N interpretation of the codes for calculating thresholds, rather than the Fibonacci number system used in [Ostromoukhov et al. 2004]. As for the algorithm of adaptive importance sampling with polyominoes, it is very close to that used in the cited article. Pseudo-code for our algorithm is provided in Appendix A.

Let us present the rationale for using polyominoes. *Polyomino* or *n-omino* is a plane topological disc, consisting of n edge-to-edge adjacent squares. Figure 2-top shows a few simple polyominoes. Polyominoes and their properties have been extensively studied in mathematics, and more precisely in combinatorial geometry [Golomb 1996; Grünbaum and Shephard 1986; Clarke 2006]. A typical problem related to polyominoes can be formulated as follows: determine whether a given planar polygon can be filled, with no gaps, by a given set of polyominoes. For example, Figure 2-bottom-left shows how the rectangle of dimensions 12×9 can be filled with 18 identical G-hexominoes.

In this paper, we consider a special class of *rectifiable polyominoes*. A polyomino is said to be "rectifiable" if several copies of the polyomino form a rectangle. Rectifiable polyominoes can always be presented in terms of self-similar $\mathscr{L}^2$-rep constructions (also called *production rules*), where the larger version of the polyomino is built out of $\mathscr{L}^2$ identical copies of the polyomino. Here, $\mathscr{L}$ is the linear scaling factor in the $\mathscr{L}^2$-rep construction, and $\mathscr{A} = \mathscr{L}^2$ is the area scaling factor. An example of a production rule for decomposition of the 9^2-rep G-hexomino into 9^2 identical G-hexominoes is shown in Figure 2-bottom-right (in this case $\mathscr{L} = 9$, and $\mathscr{A} = \mathscr{L}^2 = 9^2$). Applying the production rules iteratively, and keeping the size of polyominoes constant, one can fill an arbitrarily large planar patch, and, at the limit, the entire plane [Grünbaum and Shephard 1986]. For a given rectifiable polyomino, $\mathscr{L}$ is not unique: a variety of $\mathscr{L}^2$-rep constructions, for different linear scaling factors $\mathscr{L}$, can be found. [Clarke 2006] presents the largest known collection of $\mathscr{L}^2$-rep polyominoes, sometimes called "reptiles".

In this paper, we exploit very distinct spectral properties of rectifiable polyominoes. These properties are the consequence of the non-periodic, self-similar nature of $\mathscr{L}^2$-rep polyominoes. In fact, according to Statement 10.1.1 in [Grünbaum and Shephard 1986], if a monohedral $\mathscr{L}^2$-similarity tiling has a unique production rule, then such a tiling is not periodic.

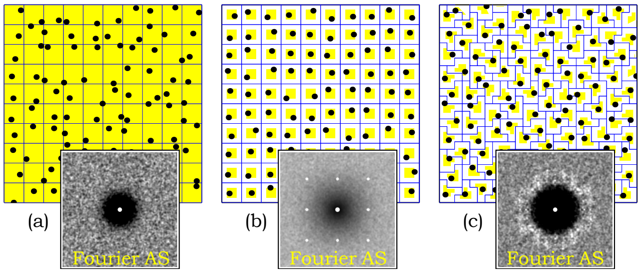


Figure 3: *(a) Stratified jitter, and its Fourier amplitude spectrum. (b) Sampling dots are restricted to the central area of each square. In this example, allowed area shown in yellow is 1/4 of the area of the square. (c) Jittering within the central area of G-hexominoes. Note that whereas the allowed area has the same ratio 1/4, its Fourier spectrum is much better.*

The sampling method presented in this paper is applicable to many $\mathscr{L}^2$-rep polyominoes. Although we have experimented with several rectifiable polyominoes, we decided to use, for the purpose of illustration, the 9^2-rep G-hexomino composed of six squares (its production rule is shown in Figure 2) which produces the best distributions we have found so far, and the 2^2-rep P-pentomino composed of five squares (production rules are shown in Figure 9) which generates a quad-tree structure, very useful in computer graphics.

One of the simplest and the most popular sampling methods used in rendering is stratified jittering. The plane is cross-ruled, and one sampling point is randomly placed per square, as shown in Figure 3a. Its Fourier spectrum shows good angular isotropy. Different studies have shown good global uniformity of the distribution. Nevertheless, there is no guarantee of good local distribution: clusters of dots or relatively large holes may appear here and there, as is clearly visible in Figure 3a. This results in relatively large noise, compared to Poisson-disc or blue-noise distributions. A naïve "improvement" of stratified jittering would be to restrict the allowed sample's position to a smaller central part of each square, shown in Figure 3b in yellow. The strong regular-grid component in its Fourier spectrum may result in severe aliasing artifacts. In contrast, placing sampling points within the central area of each G-hexomino, as shown in Figure 3c, produces much better Fourier spectra and consequently many fewer aliasing artifacts. This Fourier spectrum is indeed far from being a perfect blue-noise spectrum, but it shows clearly one important thing: *there are spatial pavements of the plane that are better than others, from the spectral point of view.* As we show in this article, applying relaxation on this initially good distribution considerably improves the spectra (see Figures 6 and 7).

We take advantage of two important properties of rectifiable polyominoes. First, their construction is simple and deterministic, and their geometrical properties can be exhaustively studied. Second, rectifiable polyominoes are fundamentally self-similar. Consequently, we can easily build a sampling system very close to the Penrose tiling-based one, but without residual peaks in the Fourier domain.

As in the Penrose tiling-based system, the key notion we need is that of *structural index*, which designates the local neighborhood of each tile. The idea is quite simple: polyominoes having identical neighborhoods, and consequently identical structural indices, will behave similarly in the process of relaxation. All useful information such as optimal position of the sampling points within each polyomino can be indexed by structural indices and referred to one reference level of subdivision or octave.

In the next section, we show how structural indices can be found for polyominoes. Then, we describe very briefly the process of optimization and ranking. In Section 4, we present a method for overcoming the limitations of deterministic sampling with polyominoes. In Section 5, we discuss some of the timing issues. Finally, in Section 6, we draw some conclusions and discuss future work related to this article.

2 Structural Indices

Our goal is to identify geometrically identical configurations around each polyomino, and somehow associate these identical configurations with the production rules. At first glance, the tiling produced with rectifiable polyominoes appears intricate and difficult to understand (see, for example, Figure 1). But, with the mind's eye, we can easily find the desired identical configurations that we call structural indices.

Polyominoes are built of adjacent squares; the square vertices form a square lattice as shown in Figure 4. Let us mark each individual tile with letters around each lattice point as shown in Figure 4. Mirrored shapes are marked differently, as shown in the inset. All we need is to walk through the tiling, identify all unique combinations of marks, and tabulate them. For example, the combination {aBCpcboh-edgf-gfed-hhcb-feig-dpkj-onml-mlon-kjdp-qcbr-saAt} corresponds to structural index 22, as shown in Figure 4 with a bold blue label (we conventionally start with the lattice point marked with 'a', then continue with the lattice point marked with 'bc', etc.; the enumeration is clockwise). This identification is rotation-invariant.

Several properties related to structural indices can be proved. First, for a given production rule, the number of structural indices is finite. For example, for the 9^2-rep G-hexomino, the number of structural indices is 436. Second, subdivision of a polyomino having a certain structural index produces a unique combination of structural indices in the subdivided configuration. Figure 4-right shows a configuration of structural indices, after subdivision of the polyomino having structural index 22. We tabulate the set of such configurations for all existing structural indices, and we call this table "structural indices production rules". This table is used in every subdivision, according to the polyomino's attribute "structural indices", which is an index to the "structural indices production rules". Thus, starting with a polyomino having any structural index, we can deterministically define all structural indices of all polyominoes, after any number of subdivisions.

3 Relaxation and Ranking

As we already mentioned, one sampling point is associated with every polyomino. Polyominoes with different structural indices have different sampling point positions, and polyominoes with identical structural indices have identical sampling point positions. The positions are expressed in terms of coefficients of two basis vectors associated with each polyomino, and relative to a conventional reference point of each polyomino.

Now, our goal is to determine a table of near-optimal positions of the sampling point positions, indexed by structural indices. We start the optimization process with the table filled with arbitrary values. Producing a large number of uniform test patches, we determine all sampling positions with the current value of the table. Performing Lloyd relaxation on sampling points, we modify the table of sampling point positions, and repeat this process iteratively, until the sampling points achieve stable positions. During optimization, we deliberately restrict sampling points within the polyominoes. This process is easy to implement, and produces very good results in a few iterations. In fact, polyominoes having similar geometrical neighborhoods behave similarly.

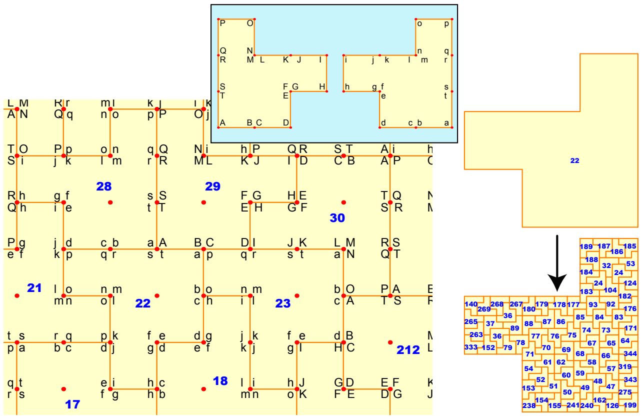


Figure 4: *(Left) Labeling pentominoes and finding structural indices. (Right) Mapping the "structural_index" attribute during the subdivision.*

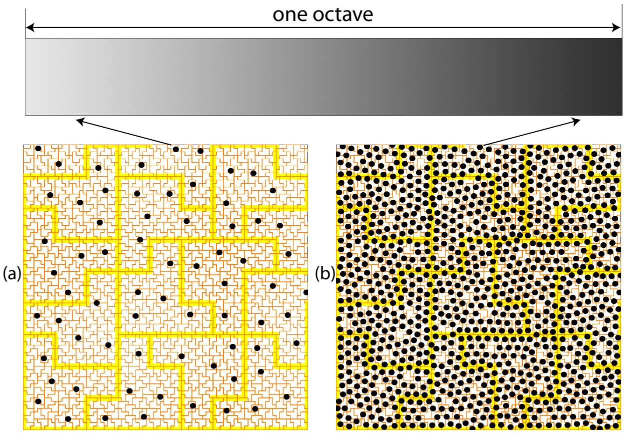


Figure 5: *Ranking, "Void-and-cluster" method: (a) at the beginning of the octave, the sampling points are placed far apart, (b) whereas at the end of the octave, the holes are far apart.*

The near-optimal solution we have obtained so far is valid only for a configuration where the area of interest is uniformly filled with polyominoes, and all polyominoes have the associated sampling points "on" or "selected". This results in discrete sample densities of $\{1, \mathscr{A}, \mathscr{A}^2, \mathscr{A}^3, \ldots\}$, which represents consecutive octaves, or levels of subdivision of the initial polyomino. To perform smooth transitions in densities within the octaves, we need to perform thresholding, which is very close to that used in Penrose tiling-based sampling. For this purpose, we maintain a special "code" attribute, similar to the "f-code" of the Penrose tiling-based sampling. All $\mathscr{A}$ polyominoes in one subdivision step are ordered. During the subdivision, an ordinal number of the tile is left-concatenated to the code. The resulting number is the threshold value; it is directly interpreted as an integer in base $\mathscr{A}$. Starting with one tile, after n subdivisions we obtain $\mathscr{A}^n$ different threshold values between 0 and $\mathscr{A}^n - 1$.

We need to determine the optimal ordering, or ranking (the term used in [Kopf et al. 2006]). We have used for ranking Ulichney's "void-and-cluster" method [Ulichney 1993], adapted for polyominoes. This greedy method is simple. Testing among all possible candidates, we select the one that produces the smallest cluster of dots, or produces the smallest hole (void). Applying this process iteratively, we can rank all $\mathscr{A}$ sub-polyominoes in one subdivision. The effect of ranking can be seen in Figure 5. At low ranks (beginning of the octave, Figure 5a), the sampling points are placed far apart, whereas at high ranks, the holes are far apart, which is the direct consequence of applying the "void-and-cluster" method. In the middle of the octave, ranking according to this method produces sub-optimal distributions of sampling points, which must be corrected in the final optimization step. The entire octave is subdivided into a number of discrete levels, and Lloyd relaxation is performed at each level. The result of this relaxation is very satisfactory (see Figure 6).

The resulting lookup table of sampling point positions is two-dimensional: one dimension is the structural index, and the other is the place within the octave for which the optimization is performed. We call the second index "importance index". Note that except for ranking, the optimization process and the structure of the table of

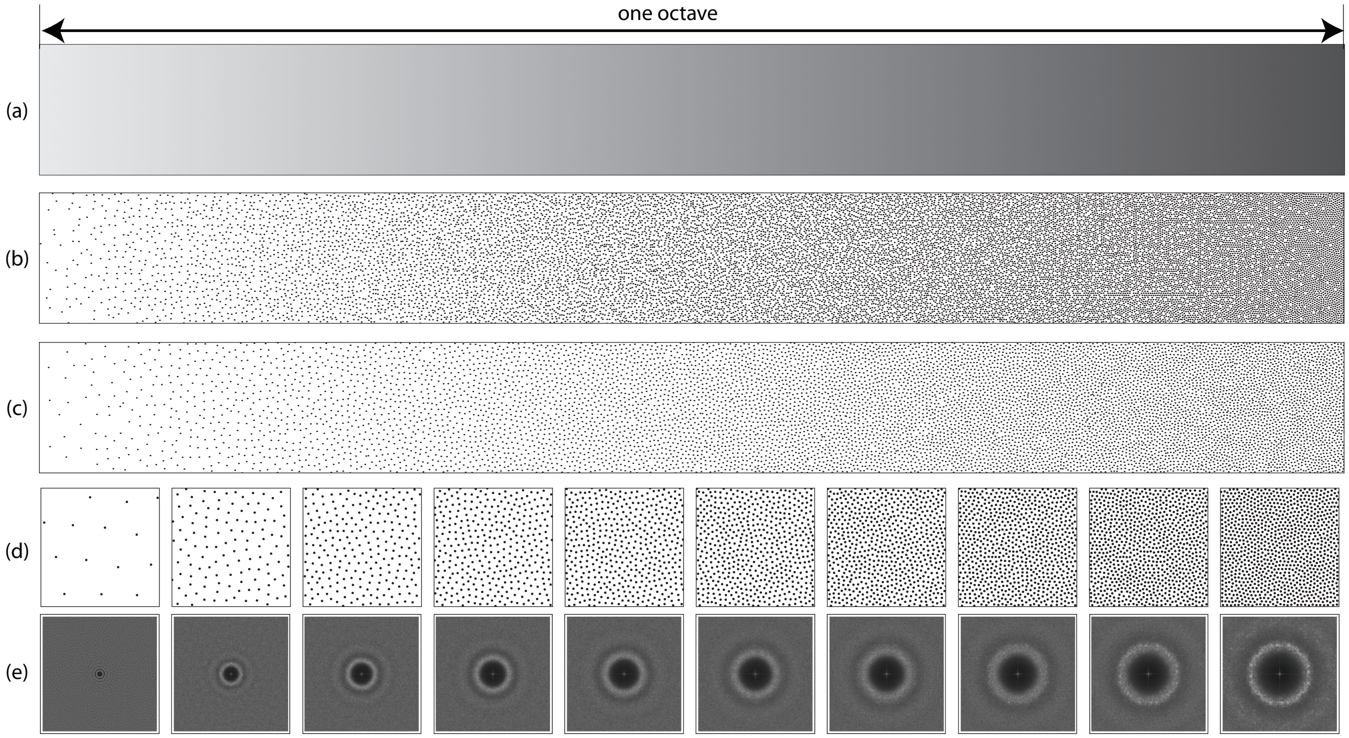


Figure 6: *Ranking and optimization. (a) One octave is shown as a smooth gray-scale importance image: importance varies according to the x axis. (b) Ranking allows generation of sampling points, which correspondxs to the importance image shown in (a). The distribution is sub-optimal in the mid-octave. (c) Sampling points, after final relaxation. Notice that this distribution is almost optimal. (d) The same octave, divided into 10 discrete sub-levels. Flat patches of constant sub-levels importance are shown, together with the corresponding Fourier amplitude spectra (e).*

sampling point positions described in this section are very close to that used in [Ostromoukhov et al. 2004].

For an arbitrary importance image, our method produces very satisfactory results, as shown in Figures 1-right and 10. Spectral analysis of the sampling distributions produced with the proposed method shows good spectral characteristics combined with low noise and nice visual aspect of the distribution (Figures 6 and 7).

We have also performed an anti-aliasing test, using the setting of [Kopf et al. 2006]. We used the zone plate test pattern $sin(x^2 + y^2)$, putting one sample per pixel, then filtering with three pixel-wide Gaussian kernel (see Figure 8). As expected, our method dramatically diminishes the artifacts of [Ostromoukhov et al. 2004]. Also, it diminishes very considerably the noise level, when compared to [Kopf et al. 2006]. This figure confirms the observations visible to the naked eye: our new method is a low-noise and low-artifact method of sampling.

4 Overcoming the Limitations of Deterministic Sampling

In rendering, deterministic sampling can be seen as a substantial limitation. A trivial modification can be applied in order to transform the proposed deterministic sampling into a pseudo-non-deterministic one. We pre-calculate a large number of variants of initial "seed" patches that cover the area of interest. To produce different non-correlated distributions, we choose randomly among the variants.

One particular polyomino, the P-pentomino, is very well-suited for this goal. We used the simplest self-similar subdivision of this polyomino, having area subdivision factor $\mathscr{A} = 2^2$. There are two variants of the subdivision scheme shown in Figure 9. A large variety of initial seed patches of square shape, formed of 20 pentominoes each, can be easily constructed.

The optimization and ranking are almost identical to that explained in the previous section. The only notable modification is in finding structural indices. With multiple subdivision schemes, pentominoes having different subdivision schemes are marked differently. For example, the yellow pentominoes in Figure 9 can be marked with "*abc*...", whereas the blue pentominoes are marked with "$\alpha\beta\gamma$...". This leads to a larger number of structural indices to be manipulated by the system, but the rest our initial method works perfectly.

This particular variant presents a number of advantages. First, it results in a simple quad-tree subdivision structure. Second, the operations of scaling and threshold calculation are reduced to bit-manipulation operations, directly supported by any processor. Finally, the "seed" patches of 20 pentominoes can be directly mapped of the surface of a sphere, as shown in Figure 1. Our preferred spherical mapping is HEALPix [Gorski et al. 2005], because of its low-distortion and Jacobian-preserving properties. Consequently, all pentominoes on each initial patch sitting on HEALPix's 12 faces (see Figure 1) are equiareal, as are all subdivided pentominoes having the same level of subdivision. The results produced with this variant are very satisfactory, both spectrally and visually.

We tested our pseudo-non-deterministic sampling with pentomi-

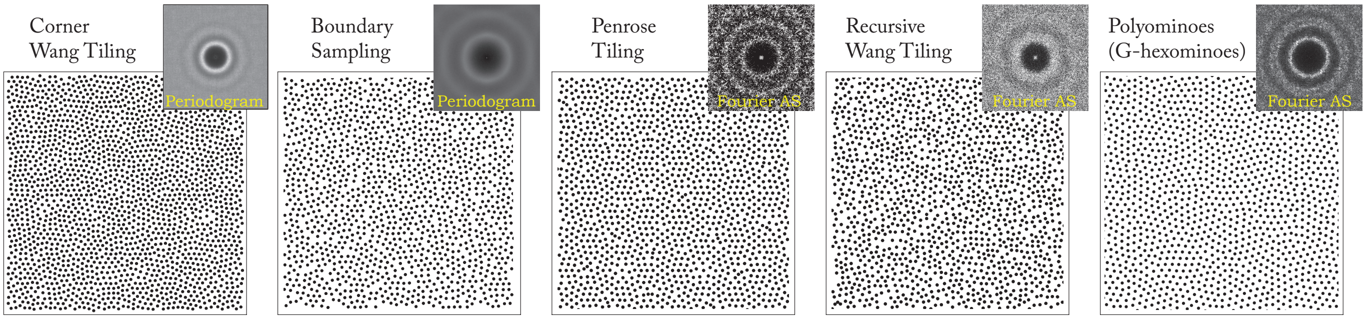
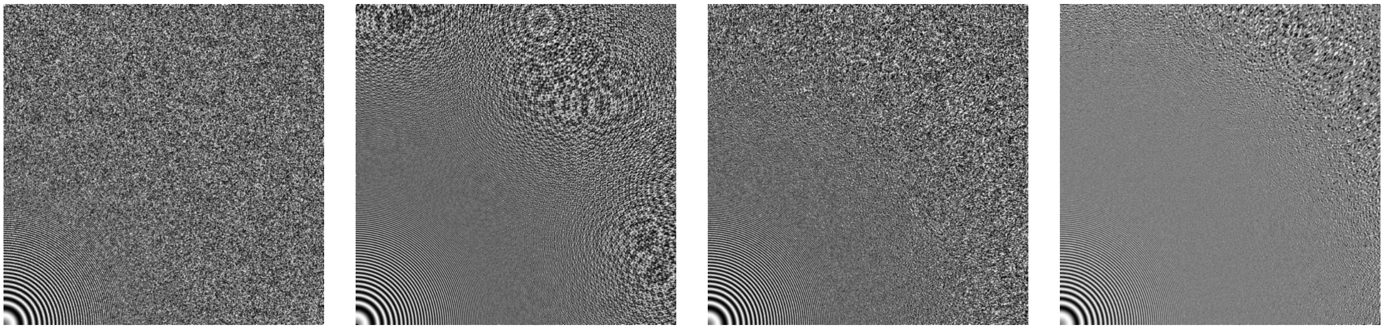


Figure 7: *Comparison with the state-of-the-art techniques for fast generation of blue-noise patterns. Note that Corner Wang Tiling and Boundary Sampling do not offer any mechanism for smooth variation of the sample density as a function of arbitrary importance.*



Jittered Grid
RMS Error: -8.1218dB

Penrose Tiling
RMS Error: -8.0873dB

Recursive Wang Tiling
RMS Error: -8.2463dB

Polyominoes
RMS Error: -8.4118dB

Figure 8: *Anti-aliasing test, using the zone plate test pattern sin*(x^2+y^2).

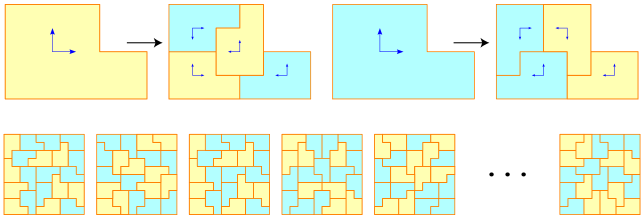


Figure 9: 2^2*-rep P-pentominoes used in pseudo-non-deterministic importance sampling. (Top) Production rules. Different colors designate different "subdivision scheme" attributes of each pentomino. (Bottom) A set of different seed patches composed of 20 pentominoes each. One patch is picked up randomly to cover the area of interest.*

noes in the context of rendering [Rousselle et al. 2007], where we perform multiple importance sampling of a complex function taking into account lighting environment, material properties, and visibility. We have observed consistent improvement over stratified jittering, in this particular context.

5 Timing

The sampling algorithm presented in this paper performs very few calculations at run time. All useful information concerning the subdivision process, such as basis vectors associated with each polyomino, positions of sampling points, etc., is tabulated. Consequently, the algorithm is extremely fast. Our timing numbers are very comparable to those mentioned in [Kopf et al. 2006]. More than one million of generated samples per second can be reached.

We would like to stress that low-level implementation details of our algorithm can be optimized. This is particularly true in the case of 2^2-rep pentominoes, presented in the previous section. In this case, all operations of scaling and threshold calculation are reduced to bit-manipulation operations. The execution time of our heavily lookup-table-dependent algorithm depends on the available hardware architecture, namely on the amount of cache memory.

6 Conclusion and Future Work

The sampling method presented in this paper offers a number of advantages, compared to the best state-of-the-art algorithms for fast importance sampling with blue noise. Our algorithm produces excellent low-noise low-artifact sampling distributions. It has been tested in the context of a complex rendering environment, and has shown no, or at worst negligible, aliasing artifacts. The run-time algorithm is almost trivial. Although the off-line optimization process is greedy and requires certain skills in manipulating non-periodic tilings, most of the people will never need to implement the optimization: C++ code together with all needed pre-calculated data structures are publicly available on our web site[1].

In the future, we would like to acquire a better understanding of the relationship between the basic polyomino's subdivision scheme

[1] `http://www.iro.umontreal.ca/~ostrom/SamplingWithPolyominoes`

and the spectral properties achieved with the optimization. As is shown in the impressive web site [Clarke 2006], the space of possible rectifiable polyominoes is unbounded. We have experimented with only a dozen subdivision schemes. A more thorough and systematic exploration of the space of all possible polyominoes would presumably provide much insight. We would like also to experiment with different optimization strategies, in order to better control the spectral properties.

Another interesting issue directly related to the present work would be importance sampling with blue noise in 3D. We are confident that many applications in computer graphics could benefit from such an efficient space sampling and partitioning algorithm with good properties. *Polycubes* [Clarke 2006] are well-suited for this purpose. This will be the topic of future research.

7 Acknowledgements

I would like to thank the anonymous SIGGRAPH reviewers. Thanks to the authors of [Kopf et al. 2006; Dunbar and Humphreys 2006; Lagae and Dutré 2006] for allowing to reproduce their images in Figures 7 and 8.

References

CLARKE, A. L., 2006. The Poly Pages. http://www.recmath.com/PolyPages.

DUNBAR, D., AND HUMPHREYS, G. 2006. A Spatial Data Structure for Fast Poisson-disk Sample Generation. *ACM Transactions on Graphics, 25*, 3, 503–508.

GOLOMB, S. W. 1996. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. Princeton University Press.

GORSKI, K. M., HIVON, E., BANDAY, A. J., WANDELT, B. D., HANSEN, F. K., REINECKE, M., AND BARTELMANN, M. 2005. HEALPix – A Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere. *The Astrophysical Journal, 622*, 759–771.

GRÜNBAUM, B., AND SHEPHARD, G. 1986. *Tilings and Patterns*. W.H. Freeman.

KOPF, J., COHEN-OR, D., DEUSSEN, O., AND LISCHINSKI, D. 2006. Recursive wang tiles for real-time blue noise. *ACM Transactions on Graphics 25*, 3, 509–518.

LAGAE, A., AND DUTRÉ, P. 2006. An Alternative for Wang Tiles: Colored Edges versus Colored Corners. *ACM Transactions on Graphics, 25*, 4, 1442–1459.

OSTROMOUKHOV, V., DONOHUE, C., AND JODOIN, P.-M. 2004. Fast Hierarchical Importance Sampling with Blue Noise Properties. *ACM Transactions on Graphics, 23*, 3, 488–495.

PHARR, M., AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann.

ROUSSELLE, F., LEBLANC, L., CLARBERG, P., OSTROMOUKHOV, V., AND POULIN, P. 2007. Hierarchical Thresholding for Efficient Sampling of the Product of All-Frequency Functions. *Submitted work*.

ULICHNEY, R. A. 1993. The Void-and-cluster Method for Dither Array Generation. In *Proceedings SPIE, Human Vision, Visual Processing, Digital Displays IV*, B. E. Rogowitz and J. P. Allebach, Eds., vol. 1913, 332–343.

APPENDIX A: Pseudo-code of the Adaptive Subdivision and Sampling with Polyominoes

```
ADAPTIVESAMP($p$ of type polyomino)
            ▷ Structure polyomino contains the fields:
            ▷   structuralIndex
            ▷   LOS: level of subdivision
            ▷   refPoint
            ▷   $v_1, v_2$: polyomino's basis vectors
            ▷   code: used for computing threshold
local_LOS ← GETMAXLOSWITHINPOLYOMINO($p$)
if $p.LOS \geq local\_LOS$
  then      ▷ Terminal: no need for more subdivisions
      local_importance ← GETLOCALIMPORTANCE($p$)
      threshold ← INTEGERINBASE$\mathscr{A}$($p.code$)
      if local_importance ≥ threshold
        then      ▷ Output selected sample
            $i_{imp}$ ← GETIMPORTANCEINDEX(local_importance)
            $\{k_1, k_2\} \leftarrow lut[p.structuralIndex, i_{imp}]$
            $position \leftarrow p.refPoint + k_1 * p.v_1 + k_2 * p.v_2$
            OUTPUTSAMPLE(position)
      return
  else      ▷ Need more subdivisions
      $\{p_1, \ldots, p_{\mathscr{A}}\}$ ← SUBDIVIDEUSINGPRODUCTIONRULES($p$)
      return {ADAPTIVESAMP($p_1$), ..., ADAPTIVESAMP($p_{\mathscr{A}}$)}
```

Routine GETIMPORTANCEINDEX() can be implemented by logirithmically subdividing each octave into *lut_size*$_2$ discrete levels:

```
GETIMPORTANCEINDEX(local_importance)
return $\lfloor lut\_size_2 * (\text{LOGBASE}\mathscr{A}(local\_importance) \bmod 1) \rfloor$
```

Linear subdivision of octaves into *lut_size*$_2$ sub-octaves is equally possible.

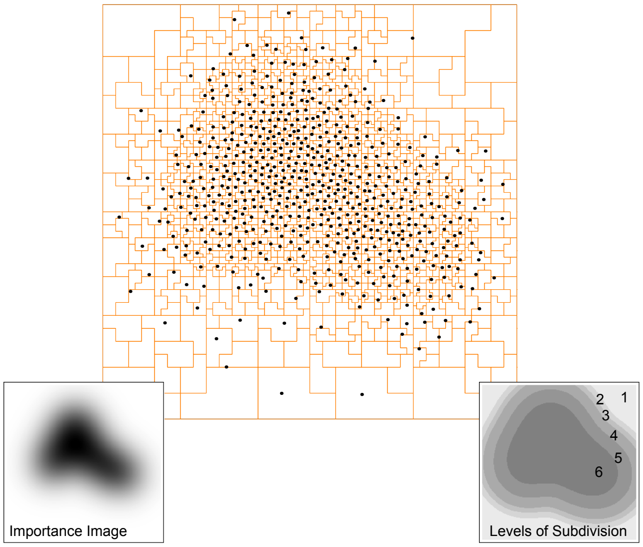


Figure 10: *An arbitrary smooth importance function, importance-sampled using 2^2-rep P-pentominoes. Notice seamless transitions between levels of subdivision.*