

Fast Tile-Based Adaptive Sampling with User-Specified Fourier Spectra

Supplementary Material

Florent Wachtel¹ Adrien Pilleboue¹ David Coeurjolly² Katherine Breeden³ Gurprit Singh¹
 Gaël Cathelin¹ Fernando de Goes⁴ Mathieu Desbrun⁴ Victor Ostromoukhov^{1,2}
¹Université de Lyon 1 ²CNRS/LIRIS ³Stanford University ⁴Caltech

In this supplementary material, we provide a more comprehensive treatment of the various technical aspects involved in our approach, along with some extensions. In particular, we show how to extend our construction (proposed in the original paper) to arbitrary subdivision factors λ and polyhex sizes m , and discuss in greater detail why we settled on our particular choices of $\lambda = 37$ and $m = 3$ based on considerations including memory size and spectral quality.

1 Tiling Construction With Randomized Boundaries

Summary: In this section, we provide formal details on the boundary randomization process. We first generalize the construction presented in our original paper to a large set of subdivision factors λ . Then, we demonstrate conditions on edge modifiers sufficient to ensure topologically consistent perturbations (Lemma 1). Finally, we detail the combinatorics of edge modifiers with respect to the desired subdivision factor (Lemma 2).

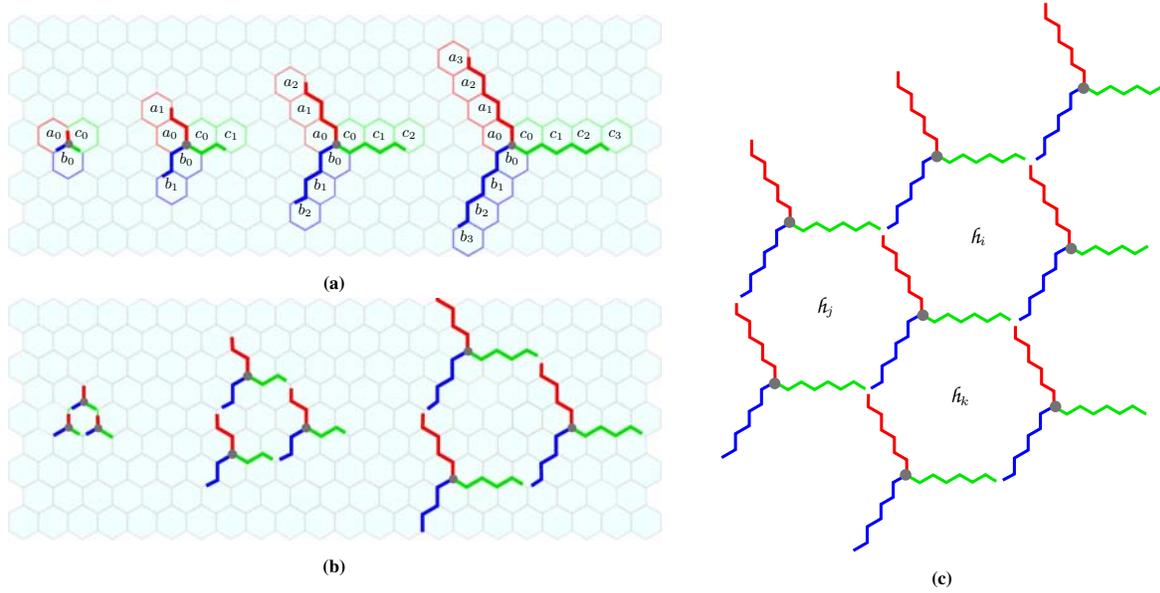


Figure 1: Construction of the regular tiling system: (a) hexagon sequences A , B and C starting from a given origin point (in gray) for $n = \{1, 2, 3, 4\}$ (from left to right). (b) tile construction via the joining of three triplets of paths for $n = \{1, 2, 3\}$. (c) overall tiling construction for $n = 4$.

From a partition of \mathbb{h} , we define a recursive structure into multi-hexagon regions we call *tiles*. Tile boundaries are comprised of hexagon sequences of equal length n aligned along one of three given directions (Fig. 1(a)):

- $A := \{a_0, \dots, a_{n-1}\}$, following the $\left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^T$ direction;
- $B := \{b_0, \dots, b_{n-1}\}$, following the $\left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)^T$ direction;
- $C := \{c_0, \dots, c_{n-1}\}$, following the $(1, 0)^T$ direction.

Triplets of A , B , and C hexagon sequences are joined at a single point, the *origin*, with hexagons a_0 , b_0 and c_0 arranged such that a_0 and b_0 are aligned along $(1, 0)^T$. These hexagons are used to define a sequence of edge paths emanating from the origin; the construction of hexagon sequences and edge paths is shown in Fig. 1(a). When three triplets of equi-length edge paths are adjoined (Fig. 1(b)), the interior forms a quasi-hexagonal region of hexagons in \mathbb{h} , i.e., a *tile*. Such tiles can be used to construct a regular paving of \mathbb{h} , denoted by \mathcal{H}^n (Fig. 1(c)). In the following, individual tiles are denoted $f_i \in \mathcal{H}^n$, with the subscript used to identify repetitions of the tile within the lattice.

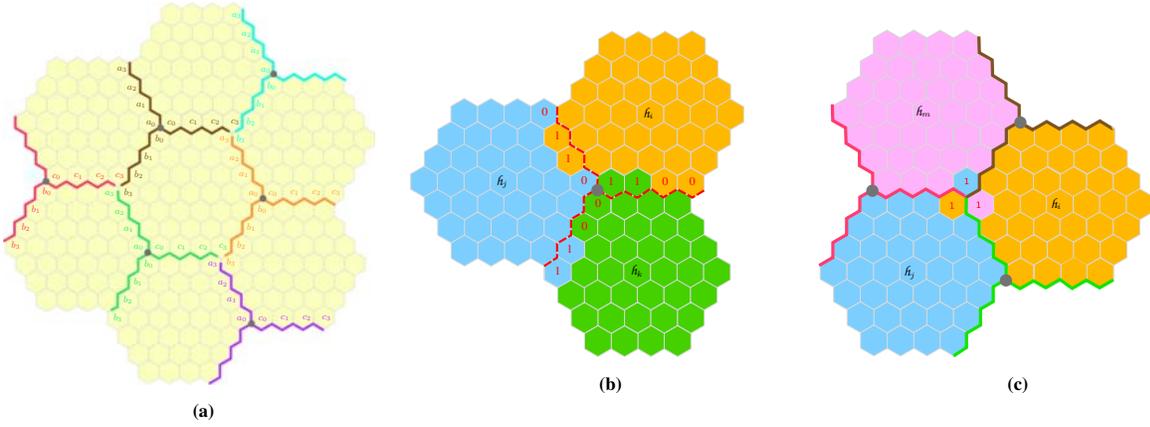


Figure 2: Edge randomization for $n = 4$: (a) subset of tiles with associated hexagons and edge words. (b) an admissible edge modification with $w_A = 0110$, $w_B = 0011$ and $w_C = 1100$. (c) illustration of a non-admissible modification with isolated hexagons: we have $w_A = 0001$ for the green triplet, $w_B = 0001$ for the brown triplet and $w_C = 0001$ for the pink triplet.

From the sequence length n , the associated subdivision factor $\lambda(n)$ of the self-similar structure is given by the number of hexagons in \tilde{h}_i . Hence, for $n = \{1, 2, 3, 4, 5\}$ we have $\lambda = \{1, 7, 19, 37, 61\}$, and more generally $\lambda(n) := C_n = 1 + 6(n-1) + 6\frac{(n-1)(n-2)}{2}$, where the sequence $\{C_n\}$ is known as the centered hexagonal number sequence [Sloane 2013c].

To perform boundary randomization, we construct irregular tiles \tilde{h}_i from tiles $h_i \in \mathcal{H}^n$ such that:

- (i) tiles $\tilde{h}_i \subset \mathbb{H}$ and $|\tilde{h}_i| = |h_i| := \lambda$, where $|\cdot|$ indicates the total number of hexagons in a tile;
- (ii) each tile \tilde{h}_i is topologically equivalent to h_i , i.e., the mapping does not generate isolated hexagons or holes.

To encode these modifications, we associate these membership changes with a binary value assigned to each hexagon along the tile boundary. Let $l : \mathbb{H} \rightarrow \{0, 1\}$ be the one-to-one map attaching the binary character to one of these hexagons. Using l , modifications to a hexagon sequence A (resp. B and C) will be characterized by a binary word of length n denoted $w_A := l(a_0)l(a_1) \dots l(a_{n-1})$, and similarly for w_B and w_C .

For example, consider changes to an A sequence encoded by w_A . If w_A contains a 0 at position i , i.e., $l(a_i) = 0$, this indicates no change along that edge. Conversely, if $l(a_i) = 1$, this indicates that the hexagon a_i is switched from h_j to h_i . Similarly, nonzero entries in w_B and w_C identify hexagons switched from h_k to h_j and h_i to h_k , respectively (Fig. 2).

Fig. 2(b) illustrates a border modification using $w_A = 0110$, $w_B = 0011$ and $w_C = 1100$. Such a triplet is called an *edge modifier triplet*. As depicted in Fig. 2(c), some edge modifiers could lead to topological issues such as isolated hexagons. We consider a set of sufficient conditions on binary words (w_A, w_B, w_C) for each triplet (A, B, C) to ensure the previously described properties on modified tiles.

Lemma 1 *The following conditions on a binary word triplet (w_A, w_B, w_C) are sufficient to obtain properties (i) and (ii):*

- (a) the number of nonzero entries in each word is equal;
- (b) words do not contain the suffix “01”.

Edge modifier triplets following these constraints are called admissible modifiers.

Proof of Lemma 1 :

Consider the binary word triplet (w_A, w_B, w_C) incident on tiles $\{h_i, h_j, h_k\} \in \mathcal{H}^n$, as shown in Figure 2. Let k_A, k_B , and k_C be the number of nonzero entries in w_A, w_B , and w_C , respectively. Following the method described previously, h_i is transformed to \tilde{h}_i by adding hexagons from h_j associated with nonzero entries in A and removing hexagons associated with nonzero entries in C . Arguments for the construction of \tilde{h}_j and \tilde{h}_k follow similarly, yielding:

$$\begin{aligned} |\tilde{h}_i| &= |h_i| + k_A - k_C = \lambda(n) + k_A - k_C, \\ |\tilde{h}_j| &= |h_j| + k_B - k_A = \lambda(n) + k_B - k_A, \\ |\tilde{h}_k| &= |h_k| + k_C - k_B = \lambda(n) + k_C - k_B. \end{aligned}$$

Hence, when $k_A = k_B = k_C$, property (i) is satisfied.

To show that condition (b) ensures property (ii), consider, without loss of generality, a nonzero value $l(a_x) \in w_A$. This indicates that hexagon $a_x \in h_j$ will be added to h_i in the construction of \tilde{h}_i . Given (b), we show that (ii) is satisfied by finding a path of adjacent hexagons from a_x to some hexagon h belonging to $h_i \cap \tilde{h}_i$. The existence of such a path demonstrates that a_x is not an isolated hexagon. Along the A sequence, h could be any hexagon save two: the first is c_0 in the the current triplet, shown in light gray in Fig. 3, which might be switched from h_i to h_k if $l(c_0) = 1$. The second hexagon, dark gray in Fig. 3, is removed from the tile if $l(\hat{b}_{n-1}) = 1$, where $(\hat{A}, \hat{B}, \hat{C})$ is the edge modifier triplet

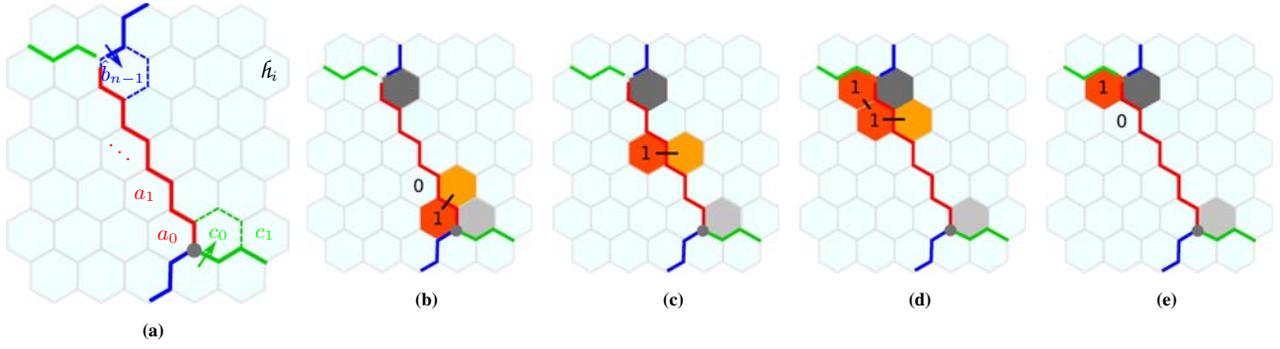


Figure 3: Illustration for Lemma 1. (a) details the local configuration. Hexagons in orange correspond to hexagon h in the text; light gray hexagons are removed from f_i if $l(c_0) = 1$; dark gray hexagons are removed if $l(b_{n-1}) = 1$. (b) case 1: a word w_A starting with “10”, (c) case 2: a word w_A containing an isolated “1” in the middle of the edge path, (d) case 3: a word w_A ending in “11”, (e) a forbidden configuration ending in “01”.

neighboring (A, B, C) across path A . Because these two hexagons may not remain in the tile when f_i is transformed to \tilde{f}_i , they cannot be used to establish topological consistency.

Observing this fact, for any $x \in [0, n - 1]$ we can specify a hexagon h in $f_i \cap \tilde{f}_i$ which is adjacent to a_x (see Fig. 3; hexagons are identified by the position of their center in the Euclidean plane):

$$\text{case 1: } x = 0, h = a_x + \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)^T,$$

$$\text{case 2: } x \in \{1, \dots, n - 2\}, h = a_x + (1, 0)^T,$$

case 3: $x = n - 1$, constraint (b) implies that $l(a_{n-2}) = 1$. Hence, a_x is adjacent to a_{n-2} , which is also adjacent to $a_{n-2} + (1, 0)^T$ under case 2.

Thus, when using edge modifiers that conform to (b), isolated hexagons will not be produced along path A . Similar arguments can be used for words w_B and w_C to show that constraining binary word triplets using conditions (a) and (b) will always ensure properties (i) and (ii). \square

In this process, the size of the set of potential tile shapes is governed by the number of admissible edge modifiers for a given edge path length n . In the general case, no closed formula exists for the number of admissible triplets, but next we will show that there is a lower bound whose asymptotic behavior is exponential in n (see Table 1 for the first few values).

| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-----------------------|---|---|----|------------|-----|------|-------|
| # Admissible triplets | 1 | 3 | 18 | 120 | 816 | 5658 | 39876 |

Table 1: Number of admissible triplets with respect to n . Numbers in bold correspond to the subdivision factor $\lambda(4) = 37$ used in the experiments presented in the original paper.

Lemma 2 Given a path of length $n > 2$, the number of admissible edge modifier triplets is given by

$$1 + (n - 1)^3 + \sum_{k=2}^n \left(\binom{n-1}{k} + \binom{n-2}{k-2} \right)^3. \quad (1)$$

A lower bound is given by the $(n - 1)^{\text{th}}$ Franel Number, $Fr(n - 1)$ (Sloane’s A000172, with $Fr(k) \approx O\left(\frac{2^{3k+1}}{k}\right)$ [Sloane 2013a]).

Proof: Let us consider a binary word of length n , with $n \geq 3$. We want to calculate the number m_k^n of binary words containing exactly k nonzero entries without containing the suffix “01”, for $k \geq 2$. These words, denoted $W_k^n = w_0 \dots w_{n-1}$, can be explicitly constructed, and form the set:

$$\mathbb{W}_k^n = \left\{ \underbrace{w_0 \dots w_{n-2}}_{\text{with } k \text{ ones}} 0, \underbrace{w_0 \dots w_{n-3}}_{\text{with } (k-2) \text{ ones}} 11 \right\}.$$

Hence, $m_k^n = |\mathbb{W}_k^n|$, and

$$m_0^n = 1, \quad (2)$$

$$m_1^n = n - 1, \quad (3)$$

$$m_k^n = \binom{n-1}{k} + \binom{n-2}{k-2} \text{ for } k \in \{2, \dots, n-1\}, \quad (4)$$

$$m_n^n = 1. \quad (5)$$

As a consequence, the number of admissible edge modifiers is given by

$$\sum_{k=0}^n (m_k^n)^3. \quad (6)$$

A lower bound for this equation is given by the first term of Eq. 6:

$$\sum_{k=0}^n (m_k^n)^3 > \sum_{k=0}^{n-1} \binom{n-1}{k}^3 \quad (7)$$

$$> Fr(n-1). \quad (8)$$

where $Fr(n-1)$ is the $(n-1)^{th}$ Franel number. \square

2 Decomposition of Irregular Trihexes into Subdivided Irregular Trihexes

Summary: This section describes the subdivision of trihex \tilde{t} into yet smaller hexagons, and the clustering of these hexagons into further trihexes (set $\tilde{\mathfrak{T}}$ in Fig. 4).

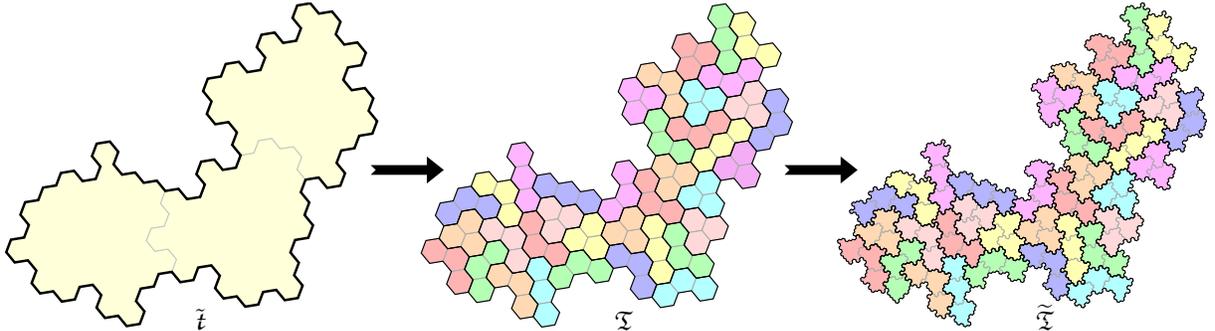


Figure 4: Decomposition of the irregular trihex \tilde{t} into regular trihexes \mathfrak{T} . Applying the subdivision system on \mathfrak{T} yields irregular trihexes $\tilde{\mathfrak{T}}$.

Given a set $\tilde{t} \subset \mathbb{H}$ of adjacent hexagons homeomorphic to a disk (Fig. 4, left), we need to subdivide \tilde{t} into disjoint trihexes, \mathfrak{T} . Following the tiling literature from discrete mathematics, it must first be determined whether \tilde{t} is tileable by the set of trihexes. Then, a valid tile decomposition for the region must be identified. In the general case—attempting to tile $\tilde{t} \subset \mathbb{H}$ with a finite set of trihexes—both problems are NP-complete [Berman et al. 1990]. Fortunately, our construction ensures that \tilde{t} is without holes, and because the number of tiles is limited (11 trihexes), we expect a better class of complexity. This is analogous to the problem of decomposing a set of unit squares into vertical and horizontal tiles of length greater than 3. This problem is NP-complete if the set is allowed holes [Berman et al. 1990] but is in P, with a linear time solution, when the set does not contain holes [Kenyon and Kenyon 1992]. Because our setting is similarly constrained, we are able to efficiently explore various valid trihex tilings of \tilde{t} . From these, we will choose the best tiling with respect to a few criteria, described below.

If a trihex decomposition exists, there will be some integer k such that $|\tilde{t}| = 3 \cdot k$. We randomly assign a label in $\{1 \dots k\}$ to each hexagon, with each label appearing exactly three times. Then, through a series of local iterations, labels are swapped between adjacent hexagons if the change minimizes the CCVT energy proposed in [Balzer et al. 2009], mimicking the discrete capacity-constrained Voronoi tessellation presented in that work¹. After several swapping steps, we check if the obtained decomposition is valid: if so, triplets of hexagons with the same label will be adjacent. Otherwise, we perform local swaps to perturb the labels of the solution and continue iterating. Minimizing the CCVT energy in this way allows us to quickly obtain valid, non trivial decompositions.

This heuristic is performed *offline* to construct the decomposition of each set \tilde{t} in the subdivision library (see original paper). In practice, only a handful of iterations are required to obtain a valid trihex decomposition. This decomposition can be optimized further using a simple penalty

¹The analogy between trihex decomposition and CCVT is clear if one defines each hexagon to have unit capacity; trihex decomposition then corresponds to tessellating into contiguous, equi-area regions, each with a total capacity of 3.

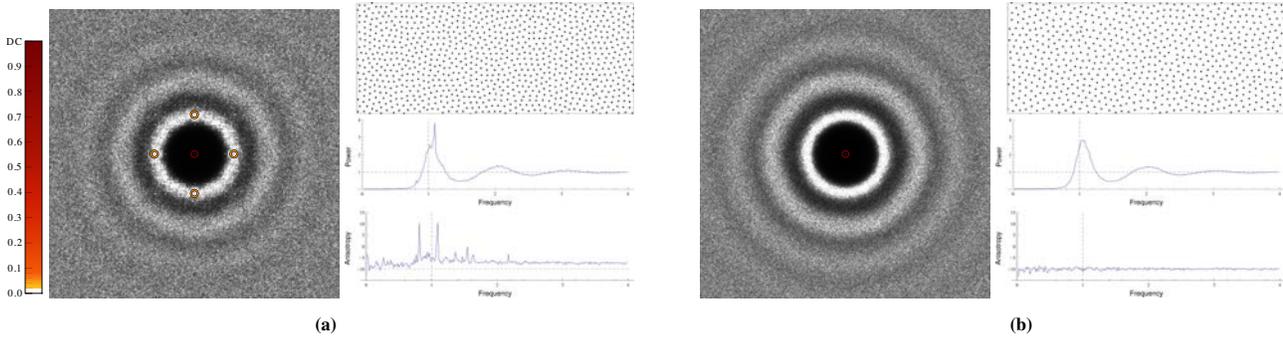


Figure 5: Comparison of spectral quality when optimizing from different initial tilings. Circles indicate values of 0.1 and greater (DC value of 1). (a) Polyomino-based tiling system [Ostromoukhov 2007] combined with BNOT optimizer [de Goes et al. 2012]. (b) Our irregular trihex tiling system with tile centroids optimized using the same BNOT optimizer.

method. For instance, we ensure that most of the trihex configurations are observed. This criteria is useful in preventing regularity from arising due to the subdivision rules. In our system, for each of the $11 \cdot \lambda$ sets requiring subdivision, we generate 100 valid decompositions. Then, we compute the distributions of trihex configurations and select the most diverse decompositions. Because the discrete CCVT minimization requires only a few iterations over small sets, the total process takes less than a minute to compute.

3 Recent optimizers applied to previous tile-based methods

Summary: Using recent optimization techniques in conjunction with previous tile-based methods gives unsatisfactory results.

In order to improve upon previous tile-based sampling methods, one might consider revising previous methods by combining them with the latest in state-of-the-art optimization techniques. Unfortunately, even the best optimization methods may be unable to alleviate the spectral peaks inherent to existing tilings. For instance, in Fig. 5 (a), we show the results produced by optimizing the centroids of polyominoes [Ostromoukhov 2007] using the method of [de Goes et al. 2012]. This distribution retains the spectral peaks found in polyomino tilings (see circled regions). In contrast, the same optimizer applied to our new tiling system produces a distribution free of spurious peaks (see Fig.5 (b)).

4 Influence of Method Parameters on Tiling Quality and Lookup Table Requirements

Summary: In this section, we explore the main parameters of our system and analyze their respective impact on the spectral quality of the tiling (see Fig. 7 and 8) and the size of the offset lookup table (see Table 2). Our sampling algorithm requires three main parameters:

- The **structural index depth**, l , affecting the size of the neighborhood referred to by each structural index;
- The **subdivision factor**, λ , of the underlying tiling;
- The **polyhex size**, m , i.e., the number of irregular hextiles comprising each polyhex.

In the following section, please refer to Fig. 7 for an illustration of the tiling system parameterized by different values for m and λ ; the corresponding spectra can be found in Fig. 8. The total number of structural indices, $|id^l|$, generated with various configurations of m , λ , and l is presented in Table 2, along with the relevant lookup table size.

Lookup table (LUT) size is simply the total number of offsets calculated for the corresponding tile set, namely: $|LUT| = |id^l| \cdot (\lambda - 1)/2$. Here, the total number of structural indices is multiplied by $(\lambda - 1)$ rather than λ , as offsets for each range of rank R_λ are equal to those for R_1 , up to a transformation. Furthermore, for each range of ranks R_r offsets are calculated only for ranks $1 \dots r$, yielding the additional factor of $1/2$: the lookup table is in fact a triangular matrix with $|id^l| \cdot (\lambda - 1)/2$ nonzero entries.

The number of structural indices is driven by two quantities: the set of l rank indices of size λ^l (see Eq. 9), and the number of neighboring configurations around each polyhex, denoted as $f(m, \lambda)$. This value is difficult to express asymptotically in m and λ ; however, it is lower-bounded by the number of hexagonal polyominoes (polyhexes) with m cells, a quantity whose behavior is exponential in m [Sloane 2013b]. The total number of polyhexes is further increased by a factor of λ due to the border shuffling step.

$$|LUT| = \left(f(m, \lambda) \cdot \lambda^l \right) \cdot \frac{\lambda - 1}{2}, \text{ with } f(m, \lambda) \equiv o(\lambda \cdot \exp(m)).$$

Structural index depth l . The structural index (Eq. 2) introduced in Sect. 3.3 of the submitted paper can be generalized as follows. Rather than characterizing the entire subdivision path from t_0 to t_{k-1} with the rank of the final subdivision step, one can instead utilize ranks for each of the last l subdivision steps. This sequence of l ranks is then appended to the neighborhood index summarizing the trihexes surrounding \tilde{t}_{k-l} , denoted by $n(\tilde{t}_{k-l})$. The resulting structural index is:

$$id^l(\tilde{t}_k) := n(\tilde{t}_{k-l}) \oplus \text{rank}(\tilde{t}_{k-l+1} | \tilde{t}_{k-l} \rightarrow \tilde{\mathfrak{X}}_{k-l+1}) \oplus \dots \oplus \text{rank}(\tilde{t}_k | \tilde{t}_{k-1} \rightarrow \tilde{\mathfrak{X}}_k). \quad (9)$$

Employing larger or smaller values of l will widen or constrict the neighborhood described by each structural index, respectively. For example, if $l = 0$, the index characterizes the first ring neighborhood (≈ 7 polyhexes); for $l = 1$, it describes 7λ neighboring polyhexes; and for $l = 2$ it describes $7\lambda^2$ neighboring polyhexes. Hence, the larger the structural index depth, the more accurate the optimization process will be (Fig. 6). Indeed, increasing the depth improves the characterization of local trihex neighborhoods and thus the support on which the point set optimization is performed. In Fig. 6, we first observe an important improvement on the spectrum and sample distribution from $l = 0$ to $l = 1$. We also observe that changes when increasing l from 1 to 2 are limited.

Note that this improvement comes at a cost: as l increases, so does the number of structural indices generated—exponentially (i.e., as λ^l). As noted above, this has direct impact on the size of the lookup table. In our implementation, we find $l = 1$ to provide a sufficient neighborhood characterization at an acceptable cost.

Subdivision factor λ . The subdivision factor determines the number of smaller hexagons into which each larger hexagon is divided. If λ is too small there will not be enough variety in the tile set to overcome the limitations common to previous tile-based sampling methods. Indeed, when hexagons are divided into a great many smaller hexagons, more highly irregular hextile borders are possible. On the other hand, finer adjustments to the border reduce the influence of each shuffled hexagon, leading to smaller magnitude perturbations of polyhex centroids. Larger values of λ also increase the size of the resulting lookup table (see Table 2).

To limit the size of the lookup table and maximize the effect of the border shuffling, our implementation uses $\lambda = 37$.

| | | Subdivision factor λ | | | |
|------------------|---|------------------------------|----------------|----------------------|---------------|
| | | 19 | 37 | 61 | |
| Polyhex size m | 2 | $ id^0 $ | 3.49k (31.4k) | 10.6k (191k) | 24.8k (744M) |
| | | $ id^1 $ | 66.2k (595k) | 39.3k (707k) | 1.51M (45.3M) |
| | | $ id^2 $ | 1.26M (11.3M) | 14.5M (261M) | 92.4M (2.77B) |
| | 3 | $ id^0 $ | 37.4k (337k) | 138k (2.48M) | 32.9k (987M) |
| | | $ id^1 $ | 710k (6.39M) | 5.10M (91.8M) | 20.0M (600M) |
| | | $ id^2 $ | 13.5M (122M) | 189M (3.40B) | 1.22B (36.6B) |
| | 4 | $ id^0 $ | 176k (1.58M) | 721k (13.0M) | 1.88M (56.4M) |
| | | $ id^1 $ | 3.35M (30.15M) | 26.7M (480M) | 114M (4.32B) |
| | | $ id^2 $ | 63.7M (573M) | 63.7M (573M) | 6.98B (209B) |

Table 2: The size of the set of structural indices $|id^l|$ is determined by the index depth l , subdivision factor λ , and polyhex size m . The resulting number of lookup table entries is given in parentheses. The entry in bold corresponds to the parameters used in the original paper.

Polyhex size m . The number of irregular hextiles in each polyhex also influences the size of the lookup table. Larger polyhexes produce many more possible tile configurations, considerably affecting the total number of neighborhood indices. This effect can be seen by scanning the entries corresponding to $|id^0|$ in Table 2.

Polyhex size also influences the quality of sampling generated by tile centroids. Because our method relies upon the clustering of subdivided hexagons into polyhexes using a CCVT-based approach (Sec. 2 of the original paper), large polyhexes—say, of size $m = 500$ —will have centroids that produce very good quality blue-noise spectra, similar to those shown in [Balzer et al. 2009]. However, the effect of such large m on lookup table size makes these types of polyhexes infeasible in our framework. In comparison, polyhexes of insufficient size ($m = 1$ or 2) produce centroid distributions with inferior blue-noise spectra.

In our implementation, we thus use polyhexes of size $m = 3$, also known as trihexes.

References

- BALZER, M., SCHLÖMER, T., AND DEUSSEN, O. 2009. Capacity-constrained point distributions: A variant of Lloyd’s method. *ACM Trans. Graph.* 28, 3, 86:1–8.
- BERMAN, F., JOHNSON, D. S., LEIGHTON, F. T., SHOR, P. W., AND SNYDER, L. 1990. Generalized planar matching. *J. Algorithms* 11, 2, 153–184.
- DE GOES, F., BREEDEN, K., OSTROMOUKHOV, V., AND DESBRUN, M. 2012. Blue noise through optimal transport. *ACM Trans. Graph.* 31, 6, 171:1–171:10.
- KENYON, C., AND KENYON, R. 1992. Tiling a polygon with rectangles. In *FOCS*, IEEE Computer Society, 610–619.
- OSTROMOUKHOV, V. 2007. Sampling with polyominoes. *ACM Trans. Graph.* 26, 3, 78:1–78:6.
- SLOANE, N. J. A. 2013. *The On-Line Encyclopedia of Integer Sequences*, A000172. <http://oeis.org/A000172>.
- SLOANE, N. J. A. 2013. *The On-Line Encyclopedia of Integer Sequences*, A001207. <https://oeis.org/A001207>.

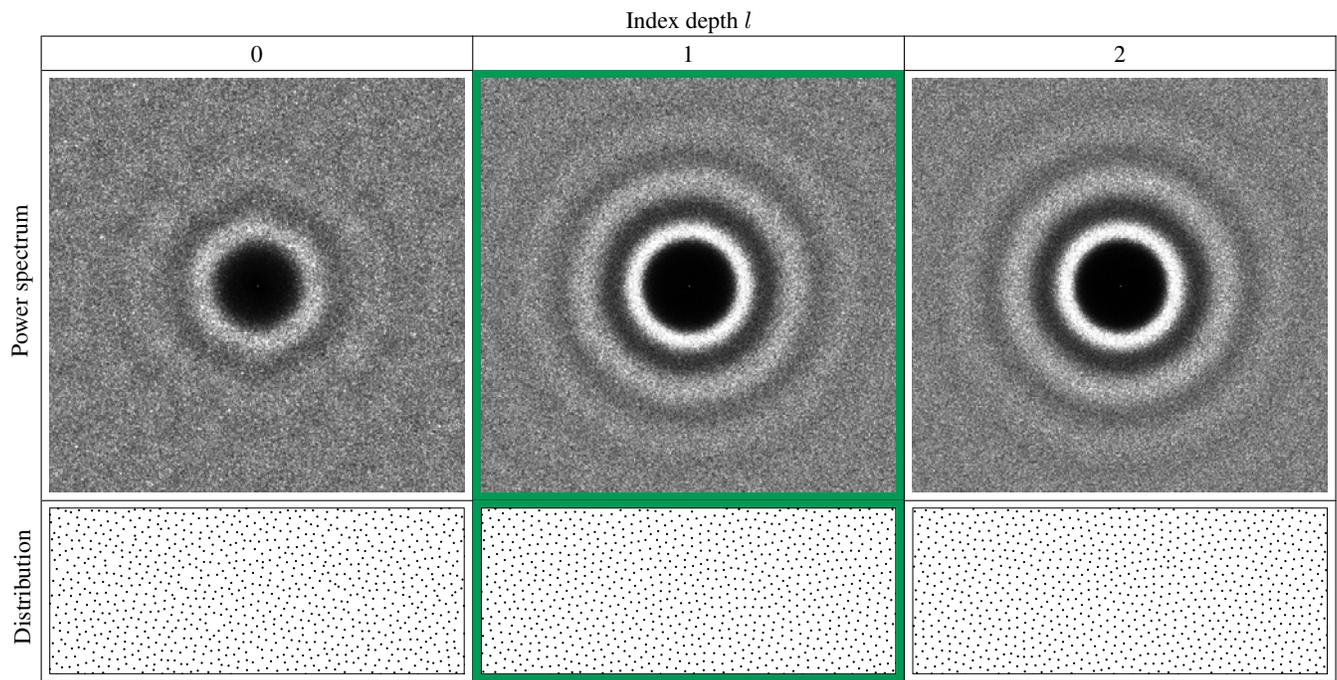


Figure 6: Spectral quality after performing our optimization, with varying index depth l as described in Eq. 9. The distribution shown is that of the range of rank R_2 . Our original paper uses $l = 1$ (green outline).

SLOANE, N. J. A. 2013. The On-Line Encyclopedia of Integer Sequences, A003215. <http://oeis.org/A003215>.

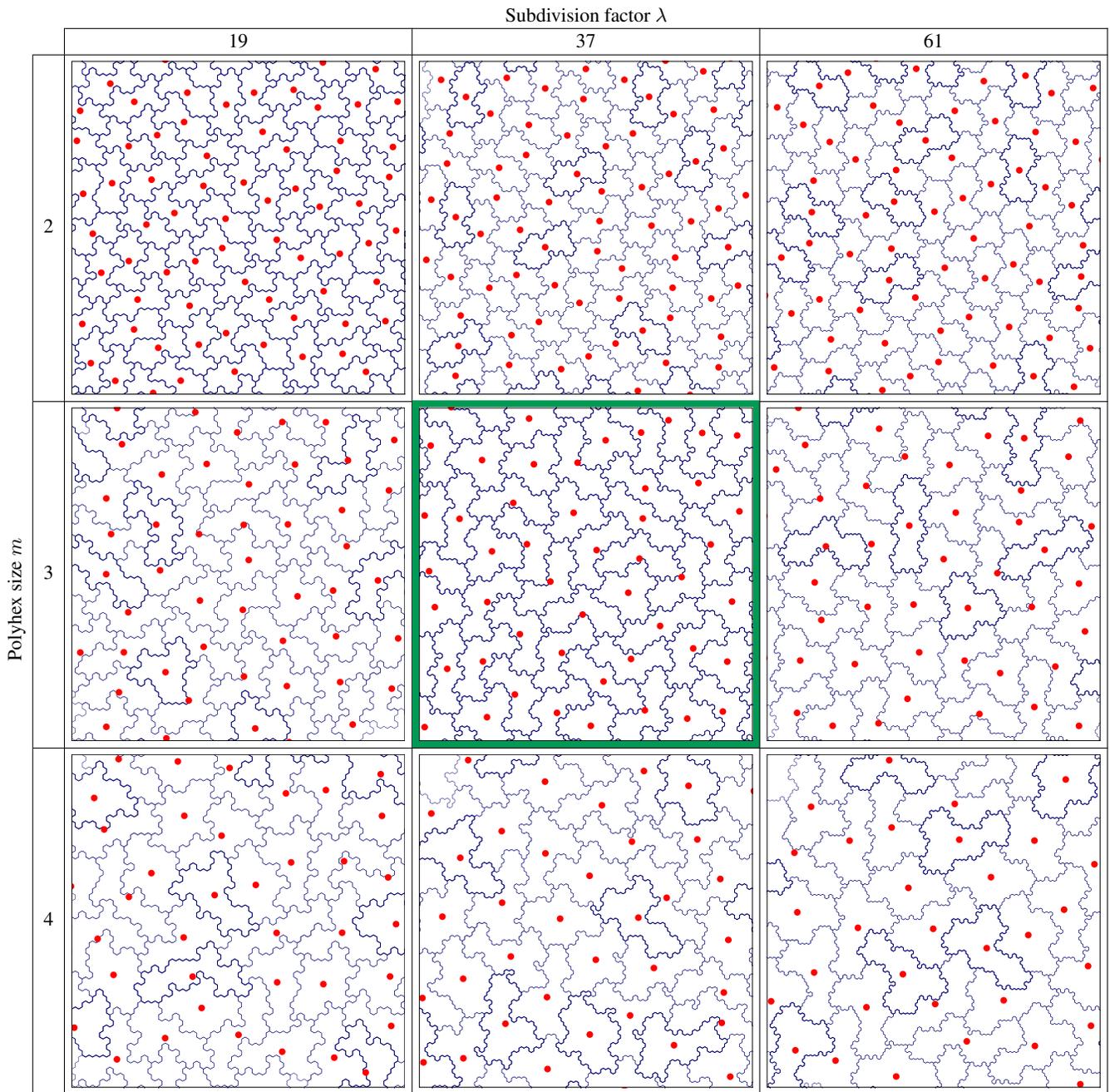


Figure 7: Tile configurations produced by varying subdivision factor λ and polyhex size. Our original paper uses the parameters shown at the center (green outline, $m = 3, \lambda = 37$).

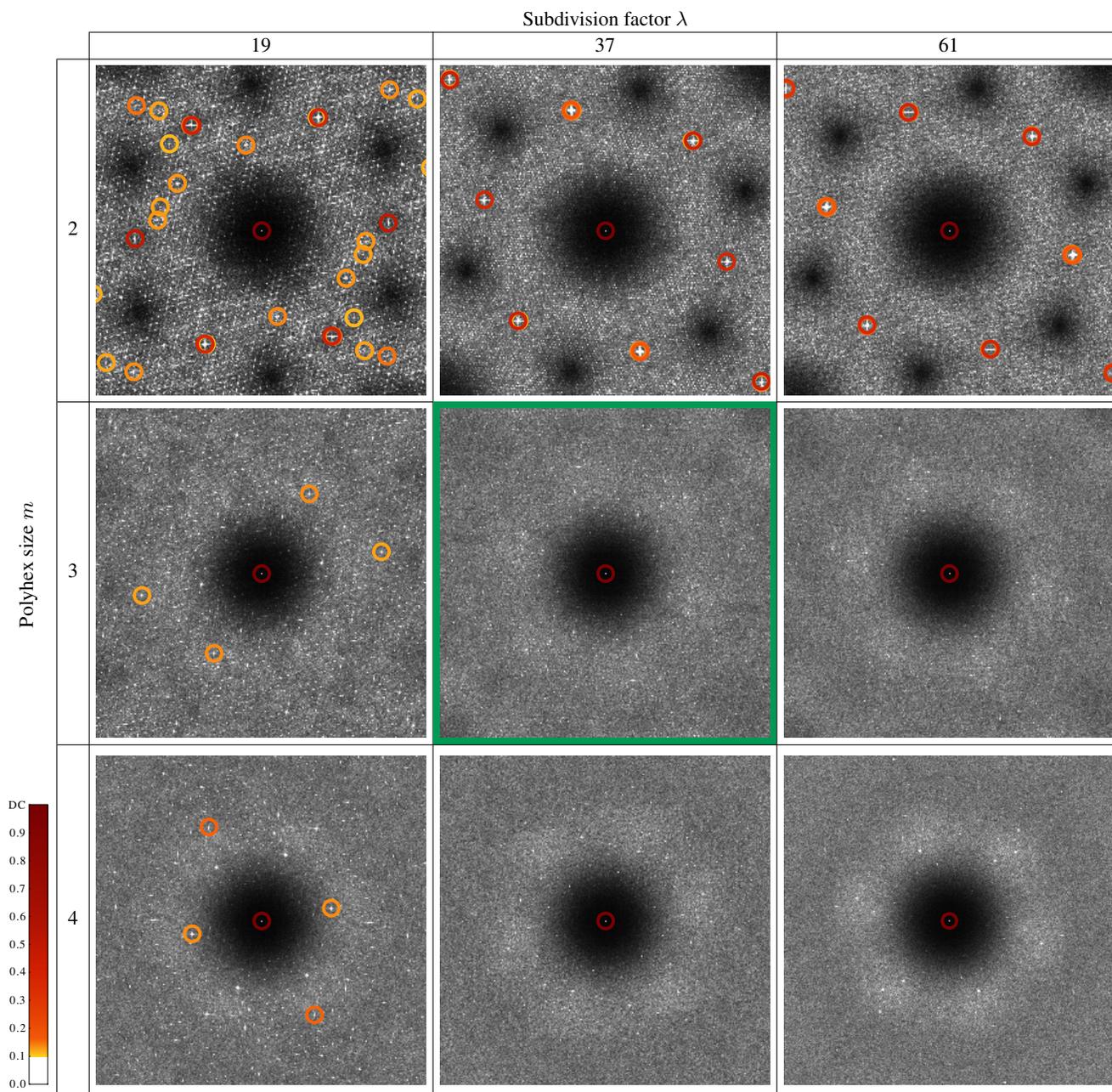


Figure 8: Spectra of tile centroids for various subdivision factors λ and polyhex sizes m . Circled marks indicate peaks of magnitude 0.1 or greater (with DC value equal to 1). Our original paper uses the parameters shown at the center (green outline, $m = 3$, $\lambda = 37$).

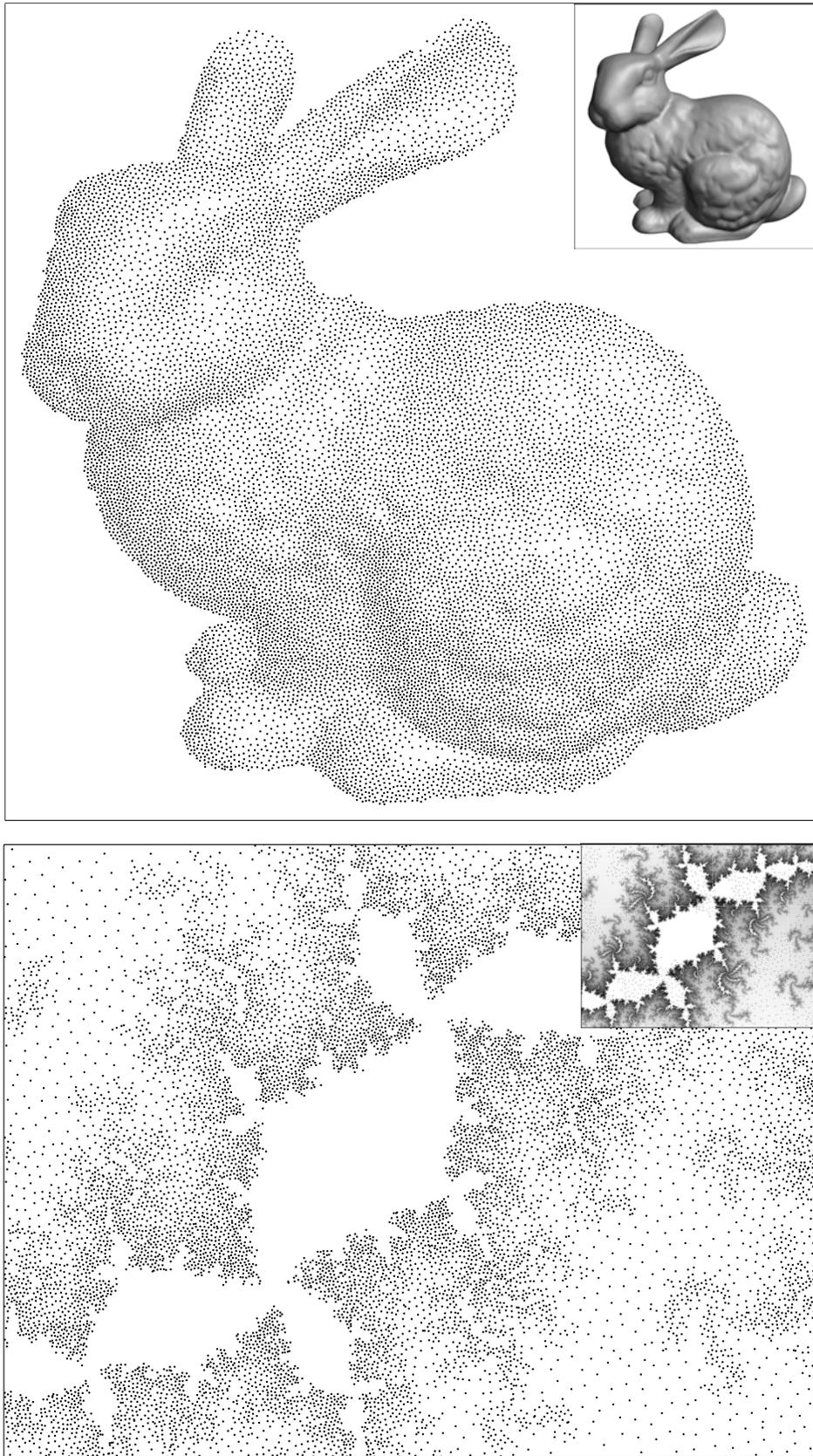


Figure 9: Realtime image sampling using our tile based method with an offline optimization based on [de Goes et al. 2012]. Top: Stanford bunny. Bottom: Julia set.