CGI2016 manuscript No.
(will be inserted by the editor)

Supplemental Material for "Multi-class anisotropic blue noise sampling for discrete element pattern generation"

Naoki Kita · Kazunori Miyata

A BuildRMatrix

In the proposed algorithm, RMatrix is constructed according to the literature [8]. In our case, the normalized RMatrix is returned by the BuildRMatrix(·) function.

Algorithm **function** BuildRMatrix($\{r_i\}_{i=0:c-1}$)

```
// user specified per-class values; {r_i},
// number of classes; c
for i = 0 to c − 1
    r ← r_i  // initialize diagonal entries
end for
sort the c classes into priority group {P_k}_{k=0:p−1} with
descreasing r_i
C ← ∅  // the set of classes already processed
D ← 0  // the density of classes already processed
for k = 0 to c − 1
    C ← C ⋃ P_k
    foreach class i ∈ P_k
        D ← D + 1/r_i^2
    end foreach
    foreach class i ∈ P_k
        foreach class j ∈ C
            if i ≠ j
                r(i, j) ← r(j, i) ← 1/√D
            end if
        end foeach
    end foreach
end for
r^scale ← r/ min(r)  // element-wise division
return r^scale
```

N. Kita
Japan Advanced Institute of Science and Technology,
E-mail: naoki-kt@jaist.ac.jp

K. Miyata
Japan Advanced Institute of Science and Technology,
E-mail: miyata@jaist.ac.jp

B Statistical Analysis for Multi-class Anisotropic Blue Noise Sampling

We analyzed our algorithm by calculating its power spectrum and anisotropy according to the literature [6]. Figures 1 to 3 show the analysis results for 3-class, 5-class, and 7-class distribution respectively. We plotted the total set and each class. To verify the anisotropic sampling, a distribution is warped back from a scaling projection to a unit square. In the original distribution, we employed a simple scaling transform from $\mathbf{p} = (x, y)$ to $\mathbf{q} = (u, v)$, i.e., $\mathbf{q} = \varphi(\mathbf{p})$, where

$$\varphi : (u, v) = (x/2, y). \tag{1}$$

Therefore, the points can be warped back by applying a Jacobian matrix expressed as

$$J(\varphi^{-1}(\mathbf{q})) = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial x}{\partial y} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

The spectrum analysis results are shown in Figs. 1−3. Here, simple scaling is used for the warp to test its anisotropic and multi-class sampling properties (left to right: original (anisotropic) samples, warped (isotropic) samples, power spectrum averaged over 10 runs, and the corresponding radial mean and anisotropy plots). The total set contains approximately 3800 samples, and each class contains almost equal number of samples.

As can be seen, the total set of all cases exhibits blue noise properties. In Fig. 1, each class also exhibits blue noise properties. However, in Figs. 2 and 3, each class is slightly biased due to a smaller number of points and may also suffer from a scaling effect and constrained distribution resulting from multi-class sampling. However, we aim for multi-class element distribution rather than a pure sampling application, such as anti-aliasing; thus, this is acceptable because we can generate visually appealing results, which are shown in the main paper.

The spectrum analysis for 16-/32-/64-classes are shown in Fig. 4. As can be seen, our algorithm can also generate distributions with blue noise properties even with a large number of classes.

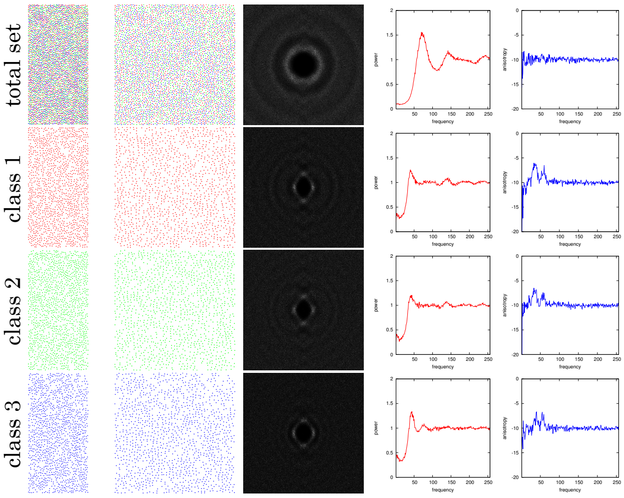


Fig. 1 Spectrum results for 3-class distribution

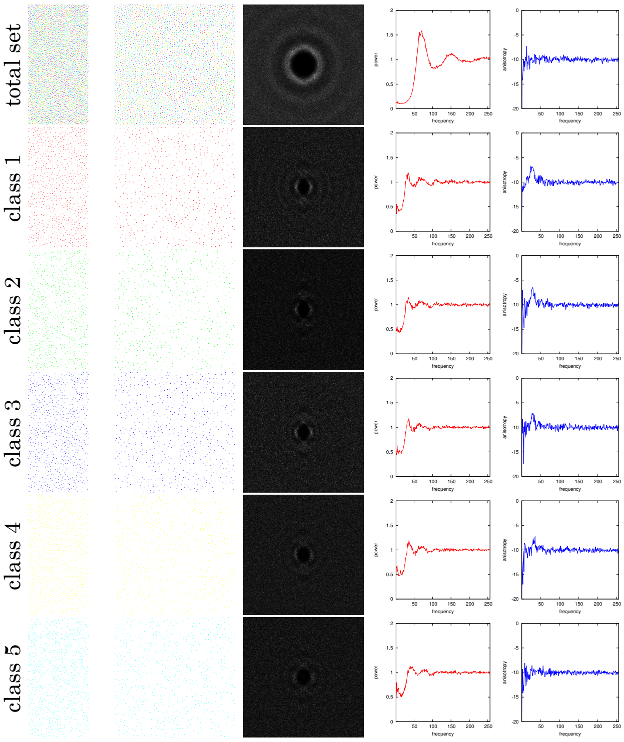


Fig. 2 Spectrum results for 5-class distribution

C Comparison of Sampling Approaches

We compare the distribution methods in the main paper. Here, we compare our method (Fig. 1) to anisotropic sampling with random class assignment (Fig. 6). We find that there are significantly sparser and denser areas of samples in each class in Fig. 6 than those shown in Fig. 1, which results in fewer blue noise qualities.

We also generate a discrete element pattern using three methods: (a) multi-class isotropic sampling [8], (b) anisotropic sampling [6] with random class assignment, and (c) our multi-class anisotropic sampling (Fig. 5). As can be seen, the anisotropy of an element is ignored and the elements are not well populated in Fig. 5a. Although Lagae and Dutré employed an

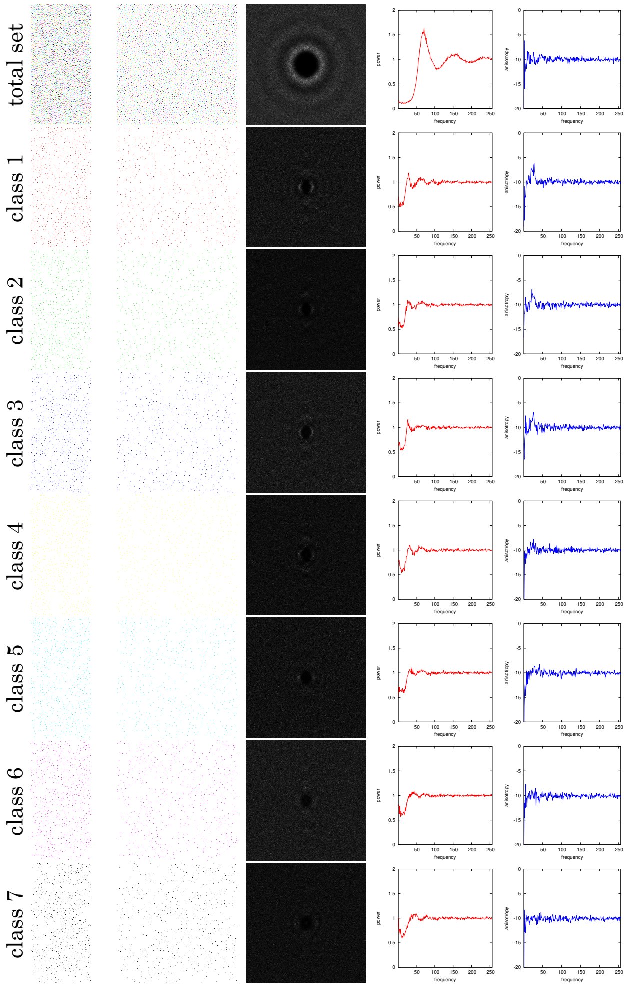


Fig. 3 Spectrum results for 7-class distribution

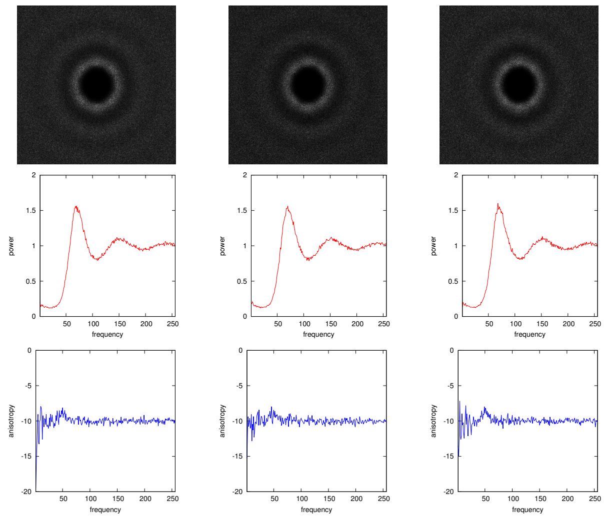
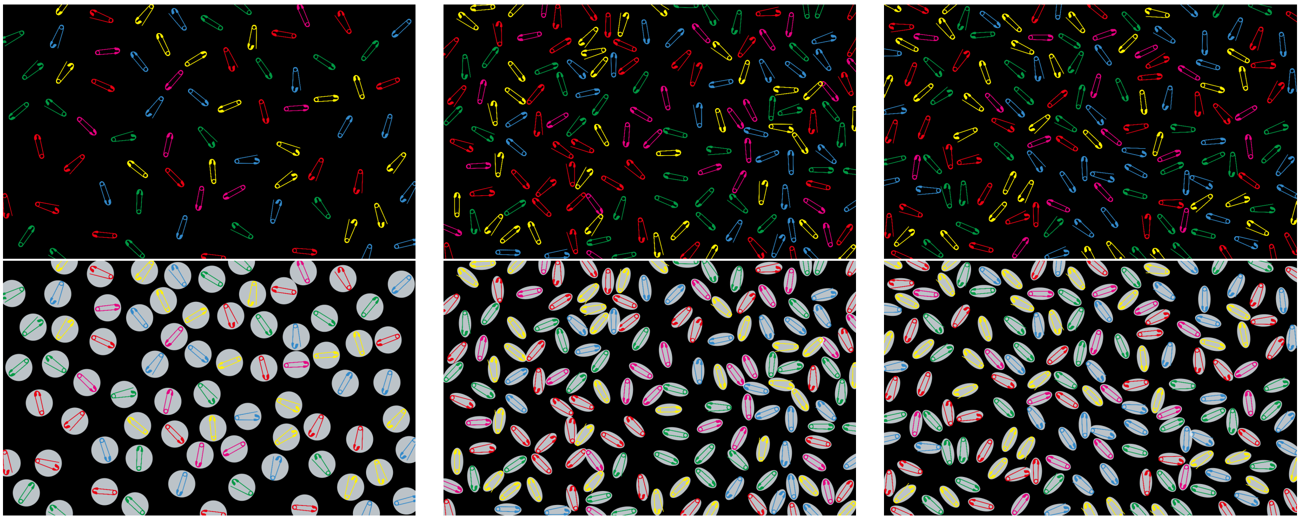


Fig. 4 Spectrum results of distribution with more classes. Left to right: 16-/32-/64-classes. In these cases, simple scaling is applied, as well as warp back from a scaling projection to a unit square for spectrum and anisotropy analysis.



(a) Multi-class (isotropic) (b) Anisotropic (random class assigned) (c) Ours

Fig. 5 Comparison of three methods: (a) multi-class isotropic sampling [8], (b) anisotropic sampling [6] with random class assignment, and (c) our multi-class anisotropic sampling. The generated patterns are shown in the top row, and the elements with proxy shapes are shown in the bottom row.

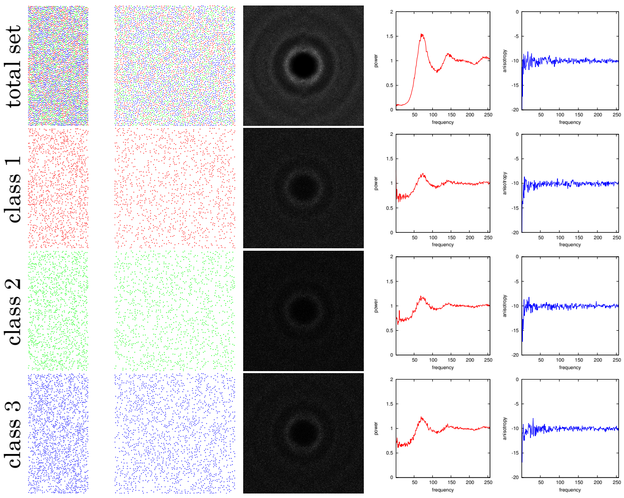


Fig. 6 Spectrum results for 3-class distribution. In this case, we first generate a distribution by anisotrpic sampling. Then, each sample is assigned a class ID randomly while maintaining a nearly equal ratio relative to the number of classes.

isotropic Poisson disk sampling approach for object distribution [4], Fig. 5 shows that isotropic Poisson disk sampling is unsuitable for anisotropic element distribution. In Fig. 5b, the elements are well populated; however, it does not consider multi-class distribution as in Fig. 5a. As a result, the same class elements are not well uniformly distributed in the whole domain. As shown in Fig. 5c, ours is well populated and the elements of the same class are uniformly distributed in the domain.

D All Patterns Shown to Participants

All of the patterns shown to the participants in our experiment are shown in Fig. 7. Since the previous discrete element placement approaches lack the ability to distribute multi-class or anisotropic elements, we compare our approach to discrete element texture synthesis approaches.

Although the statistical test shows that our method outperforms other methods, the results from these synthesis approaches are highly dependent on the input exemplar. As mentioned in Section 1 of the main paper, creating a visually appealing swatch (pattern), i.e., an exemplar, is difficult. This indicates that the proposed method might be useful while creating a visually appealing pattern, and the resulting pattern can be used as input exemplar for discrete texture synthesis approaches.

References

1. Barla, P., Breslav, S., Thollot, J., Sillion, F., Markosian, L.: Stroke pattern analysis and synthesis. In: Computer Graphics Forum (Proc. of Eurographics 2006), vol. 25 (2006). URL http://maverick.inria.fr/Publications/2006/BBTSM06
2. Hurtut, T., Landes, P.E., Thollot, J., Gousseau, Y., Drouillhet, R., Coeurjolly, J.F.: Appearance-guided synthesis of element arrangements by example. In: Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering, NPAR '09, pp. 51–60. ACM, New York, NY, USA (2009). DOI 10.1145/1572614.1572623. URL http://doi.acm.org/10.1145/1572614.1572623
3. Ijiri, T., Mech, R., Igarashi, T., Miller, G.S.P.: An example-based procedural system for element arrangement. Comput. Graph. Forum **27**(2), 429–436 (2008)
4. Lagae, A., Dutré, P.: A procedural object distribution function. ACM Trans. Graph. **24**(4), 1442–1461 (2005). DOI 10.1145/1095878.1095888. URL http://doi.acm.org/10.1145/1095878.1095888
5. Landes, P.E., Galerne, B., Hurtut, T.: A shape-aware model for discrete texture synthesis. Computer Graphics Forum (Proc. EGSR) **32** (2013)

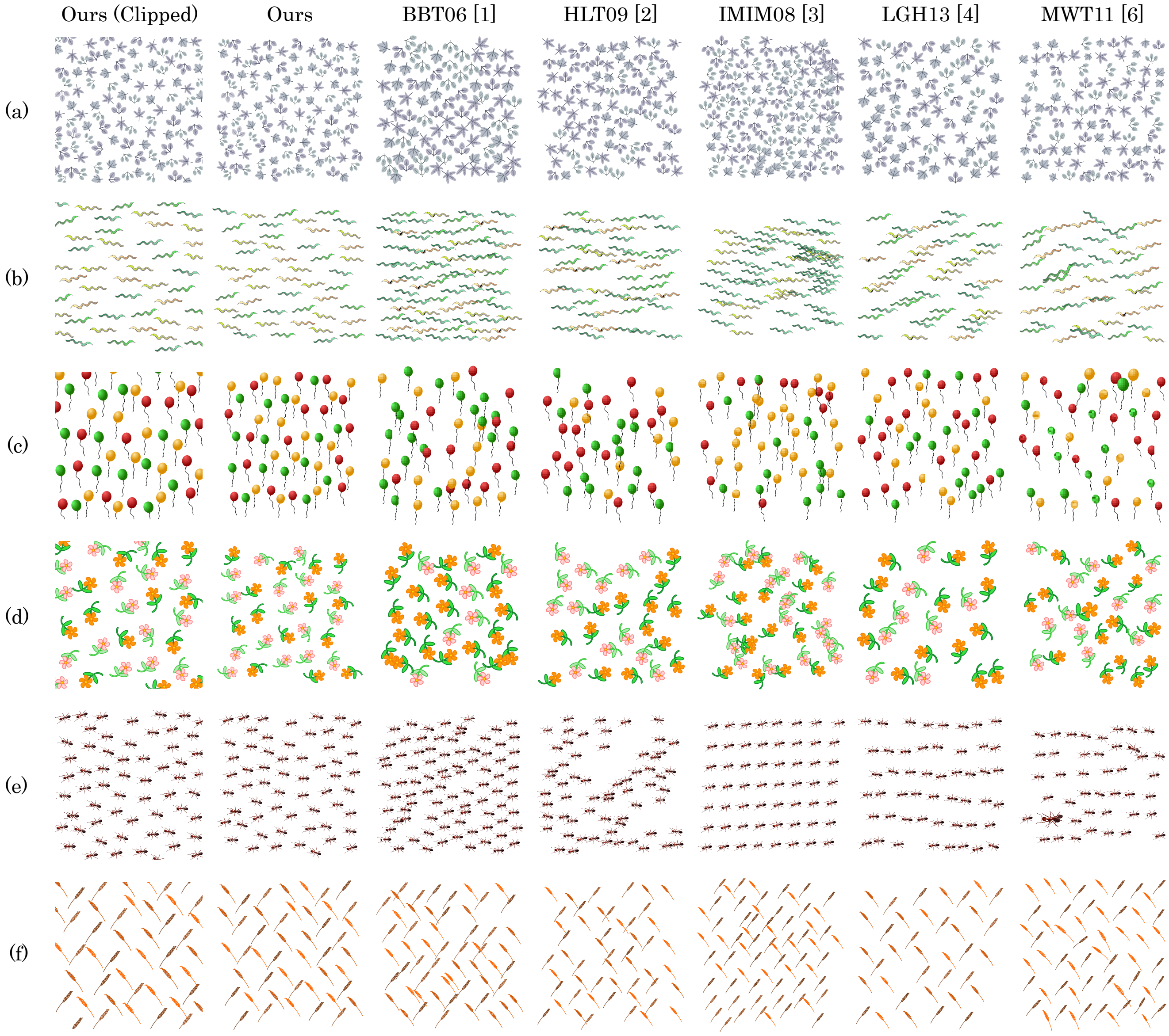


Fig. 7 All patterns shown to participants in our experiment. Left to right: ours (*clipped*), ours, BBT06 [1], HLT09 [2], IMIM08 [3], LGH13 [5], and MWT11 [7]. Top to bottom: (a) leaf, (b) snake, (c) balloon, (d) flower, (e) ant, and (f) wheat. The results other than ours are courtesy Landes et al. [5].

6. Li, H., Wei, L.Y., Sander, P.V., Fu, C.W.: Anisotropic blue noise sampling. In: ACM SIGGRAPH Asia 2010 Papers, SIGGRAPH ASIA '10, pp. 167:1–167:12. ACM, New York, NY, USA (2010). DOI 10.1145/1866158.1866189. URL http://doi.acm.org/10.1145/1866158.1866189
7. Ma, C., Wei, L.Y., Tong, X.: Discrete element textures. In: ACM SIGGRAPH 2011 Papers, SIGGRAPH '11, pp. 62:1–62:10. ACM, New York, NY, USA (2011). DOI 10.1145/1964921.1964957. URL http://doi.acm.org/10.1145/1964921.1964957
8. Wei, L.Y.: Multi-class blue noise sampling. ACM Trans. Graph. **29**(4), 79:1–79:8 (2010). DOI 10.1145/1778765.1778816. URL http://doi.acm.org/10.1145/1778765.1778816