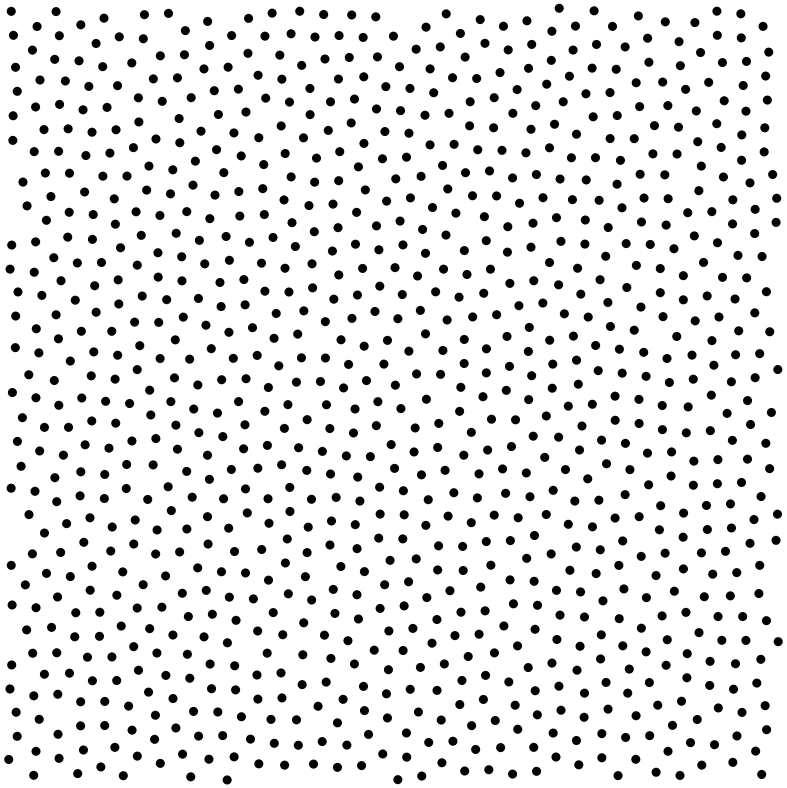
Farthest-Point Optimized Point Sets with Maximized Minimum Distance

Thomas Schlömer Daniel Heck Oliver Deussen

Computer Graphics Group
University of Konstanz

Introduction

Well-Distributed Point Sets

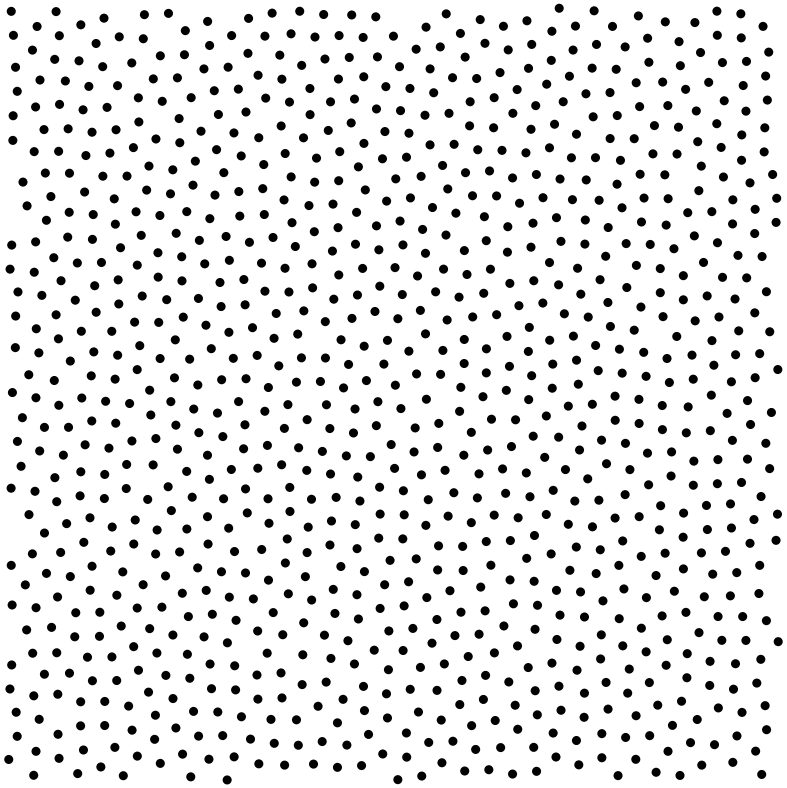


Uniform

- Avg. point density approx. constant
- No “holes” or clusters

Introduction

Well-Distributed Point Sets



Uniform

- Avg. point density approx. constant
- No “holes” or clusters

1. High minimum distance
   Prevents “clumping”
2. Irregularity
   Prevents correlations

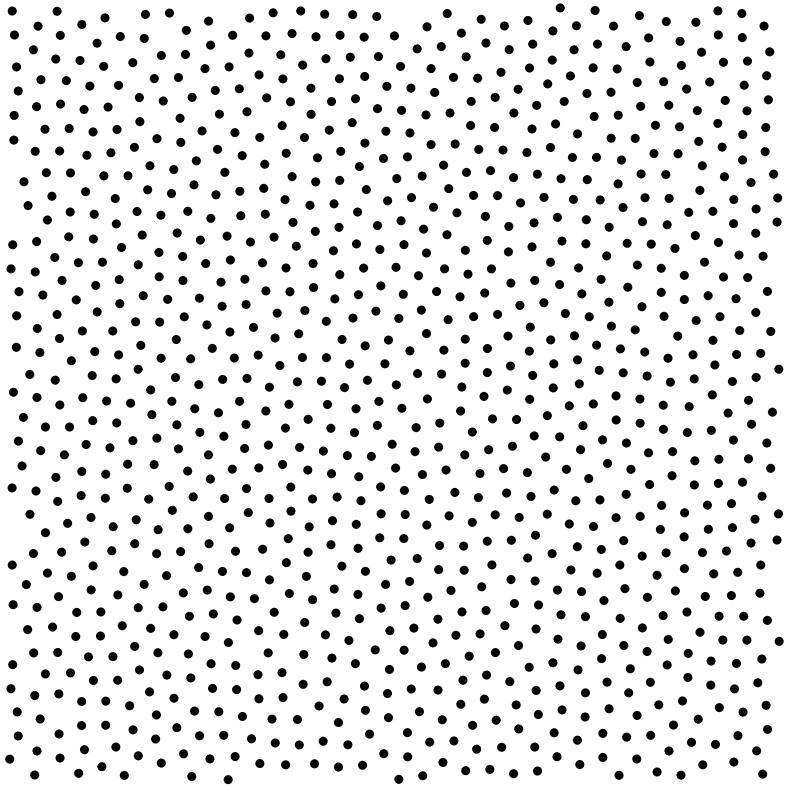
(1) and (2) combined: “blue noise”

Introduction

Introduction

1. Farthest-Point Optimization

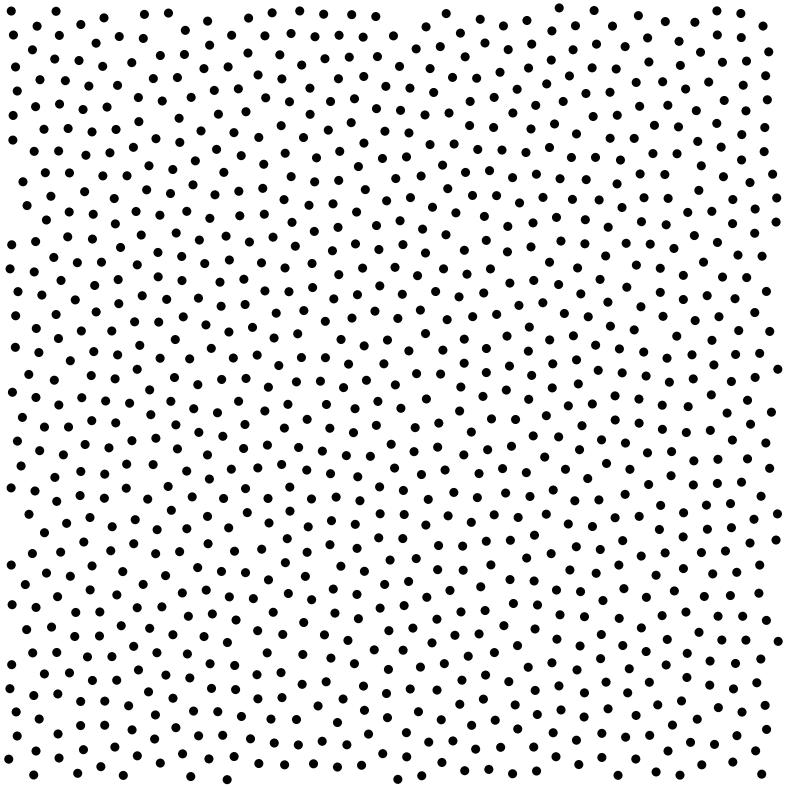
Main Algorithm



Introduction

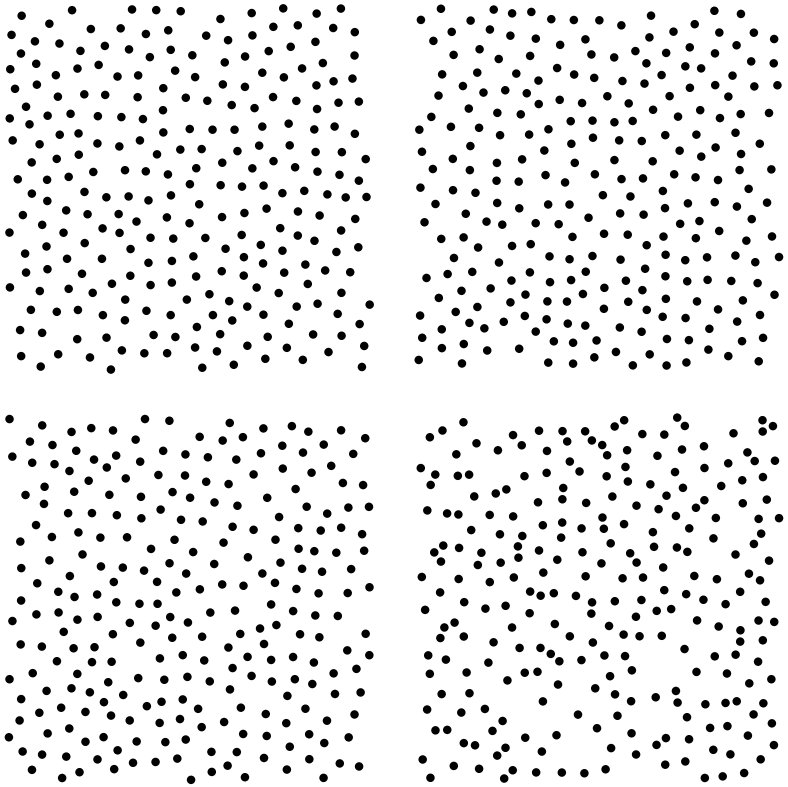
1. Farthest-Point Optimization

Main Algorithm



2. Extension

Partition Algorithm



Background

Background

Measures

- Interested in points in 2D unit square with toroidal metric $d_T(x, y)$
- For a set of points X with $n := |X|$ points we define

$$d_x := \min_{y \in X \setminus \{x\}} d_T(x, y) \qquad \text{local mindist,}$$

$$d_X := \min_{x, y \in X, x \neq y} d_T(x, y) \qquad \text{global mindist,}$$

$$\bar{d}_X := \frac{1}{n} \sum_{x \in X} d_x \qquad \text{average mindist.}$$

Background

Measures

- Interested in points in 2D unit square with toroidal metric $d_T(x, y)$
- For a set of points X with $n := |X|$ points we define

$$d_x := \min_{y \in X \setminus \{x\}} d_T(x, y) \qquad \text{local mindist,}$$

$$d_X := \min_{x, y \in X, x \neq y} d_T(x, y) \qquad \text{global mindist,}$$

$$\bar{d}_X := \frac{1}{n} \sum_{x \in X} d_x \qquad \text{average mindist.}$$

- Largest mindist by hexagonal arrangement $d_{\text{max}} = \sqrt{2/\sqrt{3}n}$
- Normalize every measure by this value

$$\delta_x := d_x / d_{\text{max}}, \quad \delta_X := d_X / d_{\text{max}}, \quad \bar{\delta}_X := \bar{d}_X / d_{\text{max}}.$$

Background

Generation Method	δ_X	$\bar{\delta}_X$	*Note*
Jittered Grid[a]	0.049	0.586	
Best Candidate[b] and FPS[c]	0.751	0.839	
Dart throwing[a] and variants[d]	0.765	0.808	
CCCVT Centroids[e]	0.778	0.896	I
CVT Centroids[f] and			
methods using Lloyd's algorithm	0.795	0.939	I, R
Electrostatic Halftoning[g]	0.826	0.952	I, R
Boundary Sampling[h]	0.829	0.862	
Low discrepancy[i]	0.903	0.920	D, R
Farthest-Point Optimization	*0.930*	*0.932*	I

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al. 1997] [d] [Lagae and Dutré 2008]
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Schmaltz et al. 2010]
[h] [Dunbar and Humphreys 2006] [i] [Grünschloß and Keller 2009]

Background

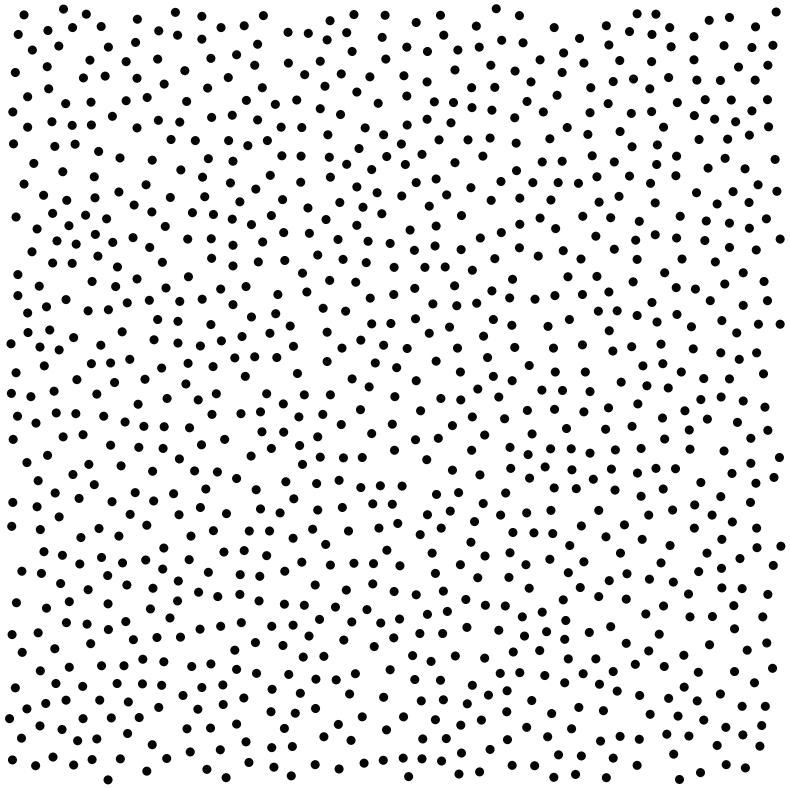
Generation Method	δ_X	$\bar{\delta}_X$	*Note*
Jittered Grid[a]	0.049	0.586	
Best Candidate[b] and FPS[c]	0.751	0.839	
Dart throwing[a] and variants[d]	0.765	0.808	
CCCVT Centroids[e]	0.778	0.896	I
CVT Centroids[f] and methods using Lloyd's algorithm	0.795	0.939	I, R
Electrostatic Halftoning[g]	0.826	0.952	I, R
Boundary Sampling[h]	0.829	0.862	
Low discrepancy[i]	0.903	0.920	D, R
Farthest-Point Optimization	*0.930*	*0.932*	I

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al. 1997] [d] [Lagae and Dutré 2008]
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Schmaltz et al. 2010]
[h] [Dunbar and Humphreys 2006] [i] [Grünschloß and Keller 2009]

Background

Generation Method
Jittered Grid[a]
Best Candidate[b] and FPS[c]
Dart throwing[a] and variants[d]
CCCVT Centroids[e]
CVT Centroids[f] and methods using Lloyd's algorithm
Electrostatic Halftoning[g]
Boundary Sampling[h]
Low discrepancy[i]
Farthest-Point Optimization

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Sc
[h] [Dunbar and Humphreys 2006] [i] [Grünschl



$\delta_X \approx 0.765,\ \bar{\delta}_X \approx 0.808$

Background

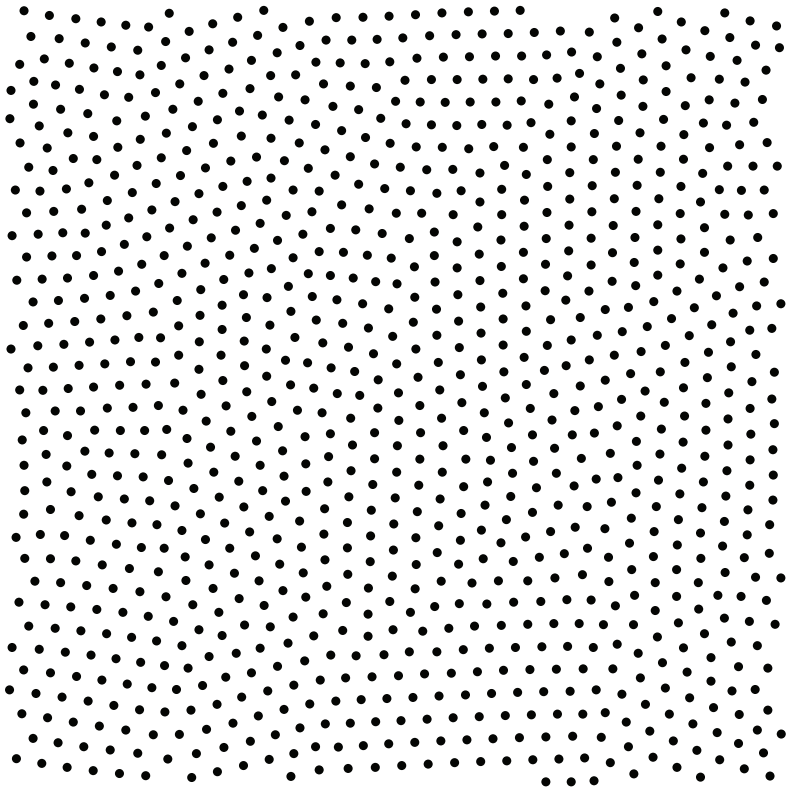
Generation Method	δ_X	$\bar{\delta}_X$	*Note*
Jittered Grid[a]	0.049	0.586	
Best Candidate[b] and FPS[c]	0.751	0.839	
Dart throwing[a] and variants[d]	0.765	0.808	
CCCVT Centroids[e]	0.778	0.896	I
CVT Centroids[f] and			
methods using Lloyd's algorithm	0.795	0.939	I, R
Electrostatic Halftoning[g]	0.826	0.952	I, R
Boundary Sampling[h]	0.829	0.862	
Low discrepancy[i]	0.903	0.920	D, R
Farthest-Point Optimization	*0.930*	*0.932*	I

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al. 1997] [d] [Lagae and Dutré 2008]
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Schmaltz et al. 2010]
[h] [Dunbar and Humphreys 2006] [i] [Grünschloß and Keller 2009]

Background

Generation Method
Jittered Grid[a]
Best Candidate[b] and FPS[c]
Dart throwing[a] and variants[d]
CCCVT Centroids[e]
CVT Centroids[f] and methods using Lloyd's algorithm
Electrostatic Halftoning[g]
Boundary Sampling[h]
Low discrepancy[i]
Farthest-Point Optimization

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Sc
[h] [Dunbar and Humphreys 2006] [i] [Grünschl



$\delta_X \approx 0.795,\ \bar{\delta}_X \approx 0.939$

Background

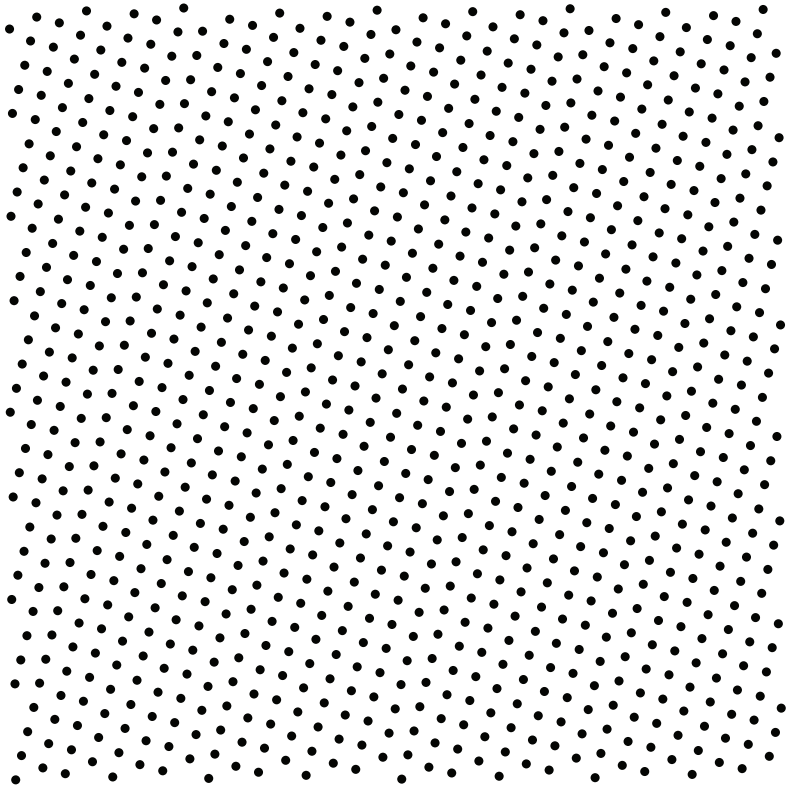
Generation Method	δ_X	$\bar{\delta}_X$	*Note*
Jittered Grid[a]	0.049	0.586	
Best Candidate[b] and FPS[c]	0.751	0.839	
Dart throwing[a] and variants[d]	0.765	0.808	
CCCVT Centroids[e]	0.778	0.896	I
CVT Centroids[f] and methods using Lloyd's algorithm	0.795	0.939	I, R
Electrostatic Halftoning[g]	0.826	0.952	I, R
Boundary Sampling[h]	0.829	0.862	
Low discrepancy[i]	0.903	0.920	D, R
Farthest-Point Optimization	*0.930*	*0.932*	I

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al. 1997] [d] [Lagae and Dutré 2008]
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Schmaltz et al. 2010]
[h] [Dunbar and Humphreys 2006] [i] [Grünschloß and Keller 2009]

Background

Generation Method
Jittered Grid[a]
Best Candidate[b] and FPS[c]
Dart throwing[a] and variants[d]
CCCVT Centroids[e]
CVT Centroids[f] and methods using Lloyd's algorithm
Electrostatic Halftoning[g]
Boundary Sampling[h]
Low discrepancy[i]
Farthest-Point Optimization

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Sc
[h] [Dunbar and Humphreys 2006] [i] [Grünschl



$\delta_X \approx 0.903,\ \bar{\delta}_X \approx 0.920$

Background

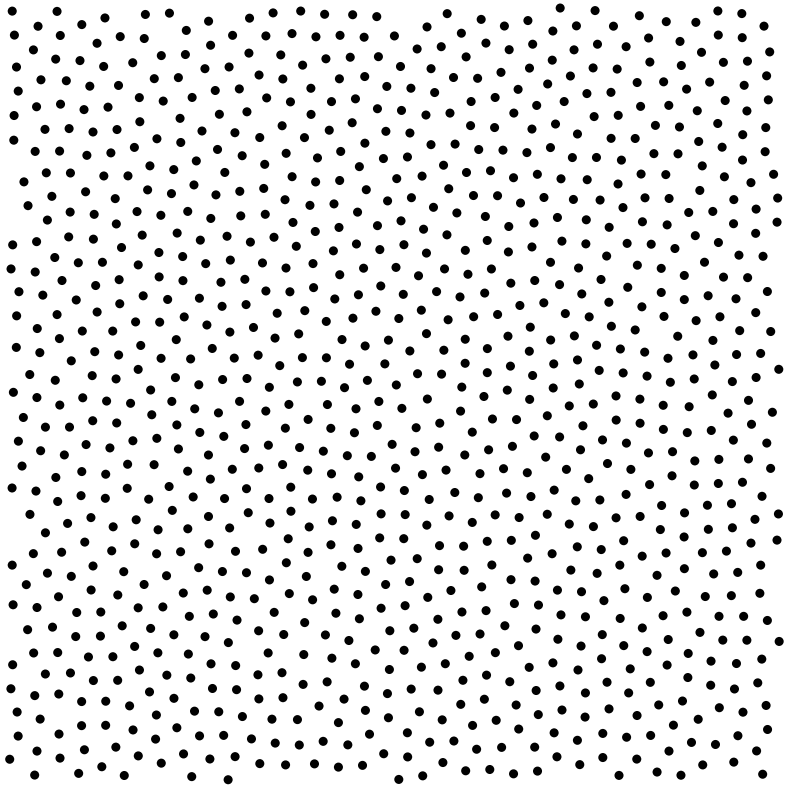
Generation Method	δ_X	$\bar{\delta}_X$	*Note*
Jittered Grid[a]	0.049	0.586	
Best Candidate[b] and FPS[c]	0.751	0.839	
Dart throwing[a] and variants[d]	0.765	0.808	
CCCVT Centroids[e]	0.778	0.896	I
CVT Centroids[f] and methods using Lloyd's algorithm	0.795	0.939	I, R
Electrostatic Halftoning[g]	0.826	0.952	I, R
Boundary Sampling[h]	0.829	0.862	
Low discrepancy[i]	0.903	0.920	D, R
Farthest-Point Optimization	*0.930*	*0.932*	I

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al. 1997] [d] [Lagae and Dutré 2008]
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Schmaltz et al. 2010]
[h] [Dunbar and Humphreys 2006] [i] [Grünschloß and Keller 2009]

Background

Generation Method
Jittered Grid[a]
Best Candidate[b] and FPS[c]
Dart throwing[a] and variants[d]
CCCVT Centroids[e]
CVT Centroids[f] and methods using Lloyd's algorithm
Electrostatic Halftoning[g]
Boundary Sampling[h]
Low discrepancy[i]
Farthest-Point Optimization

[a] [Cook 1986] [b] [Mitchell 1991] [c] [Eldar et al
[e] [Balzer et al. 2009] [f] [Du et al. 1999] [g] [Sc
[h] [Dunbar and Humphreys 2006] [i] [Grünschl

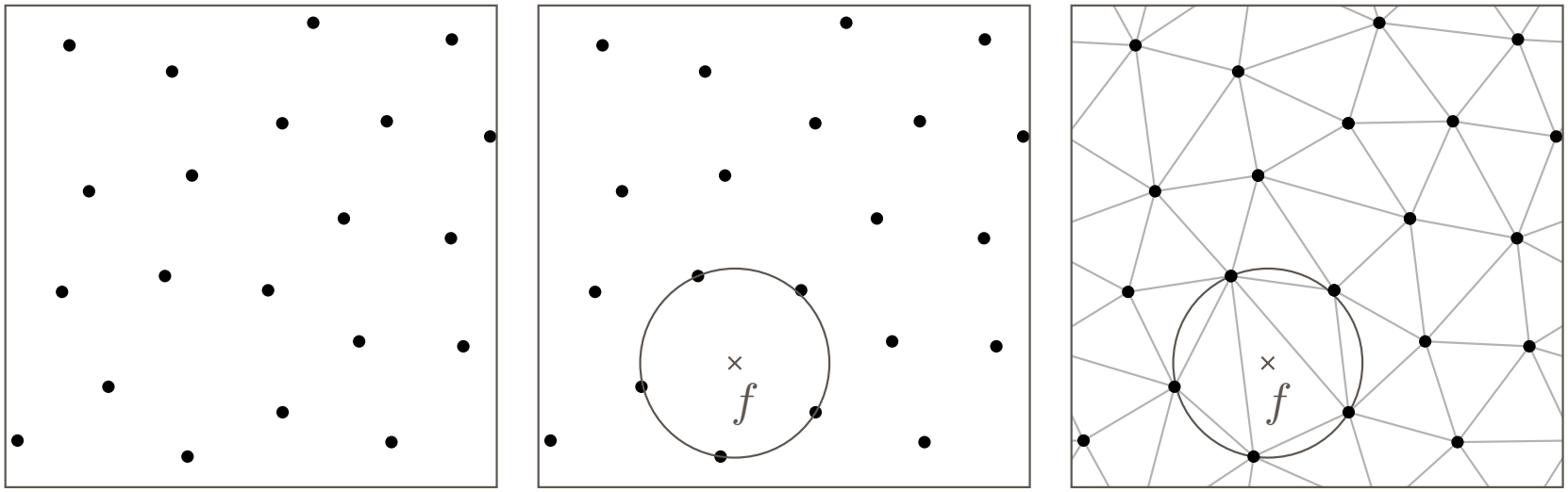


$\delta_X \approx 0.930,\ \bar{\delta}_X \approx 0.932$

Farthest-Point Optimization

Farthest-Point Optimization

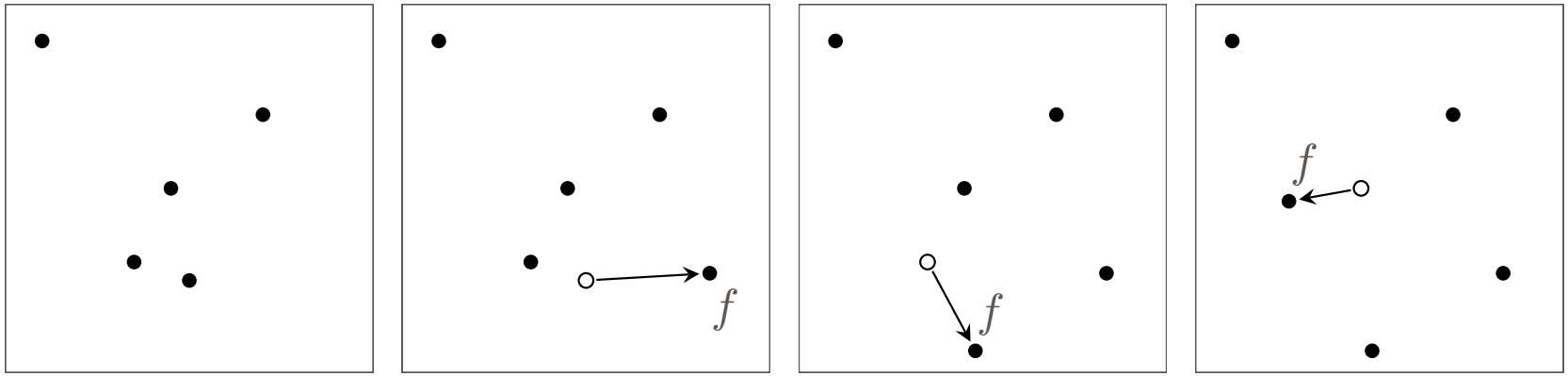
Farthest Point



- Location f that has the maximum distance from all points in X
- Corresponds to the center of the largest empty circle in the domain
- Corresponds to the center of the largest circumcircle of a triangle in the Delaunay triangulation $\mathcal{D}(X)$

Farthest-Point Optimization

Optimization Strategy



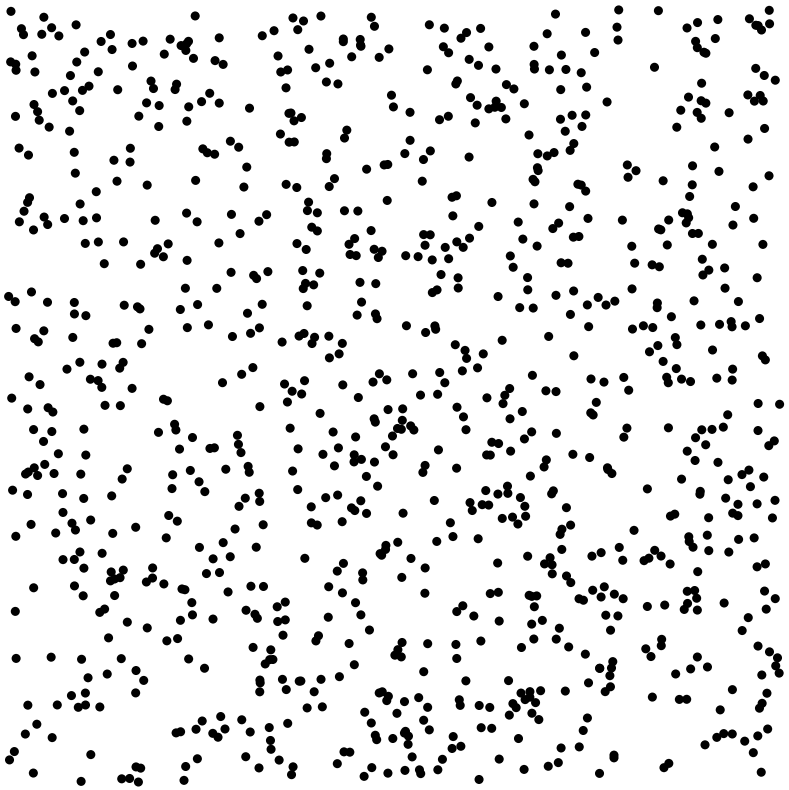
- Successively move each point to the farthest point, i.e. remove it and reinsert it at the farthest point
- One full iteration: move each point once
- Build full Delaunay triangulation once and update it dynamically

Main Algorithm

```
FARTHEST-POINT-OPTIMIZATION(X)
D = DELAUNAY(X)
repeat
    foreach vertex x in D
        (f, r_max) = (x, d_x)
        DELAUNAY-REMOVE(D, x)
        foreach triangle t in D
            (c, r) = center and radius of t's circumcircle
            if r > r_max
                (f, r_max) = (c, r)
        DELAUNAY-INSERT(D, f)
until converged
return vertices of D
```

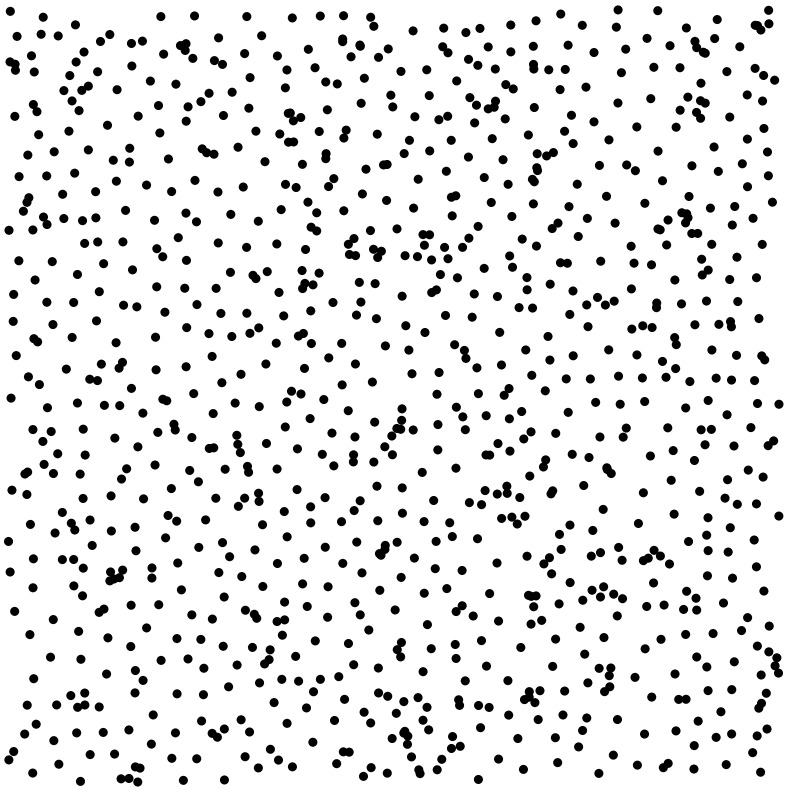
Farthest-Point Optimization

Optimizing Random Seed Points



$\delta_X \approx 0.009,\ \bar{\delta}_X \approx 0.469$

input

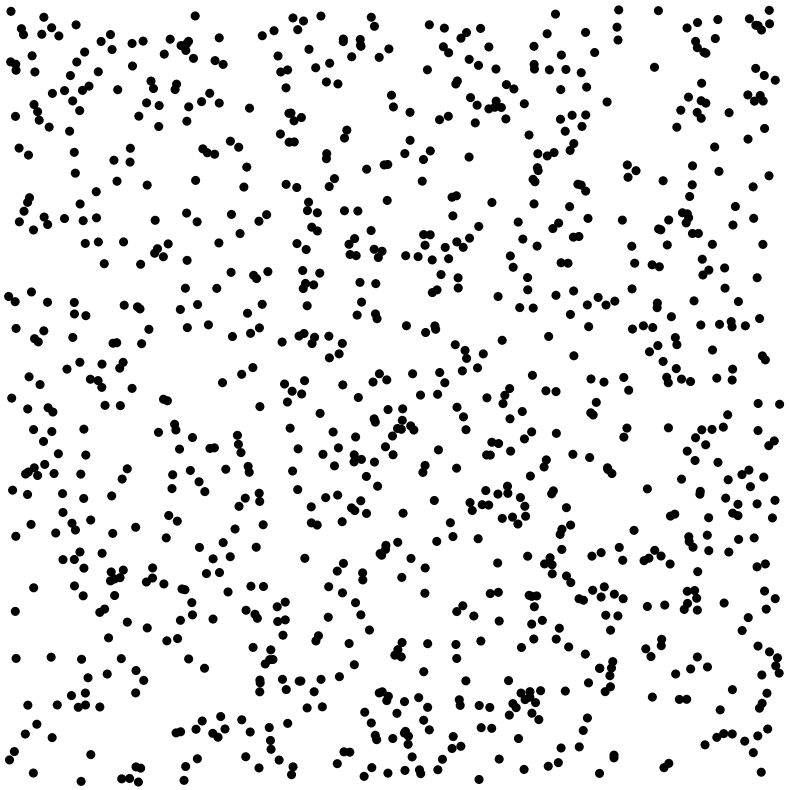


$\delta_X \approx 0.049,\ \bar{\delta}_X \approx 0.645$

1/4 iteration

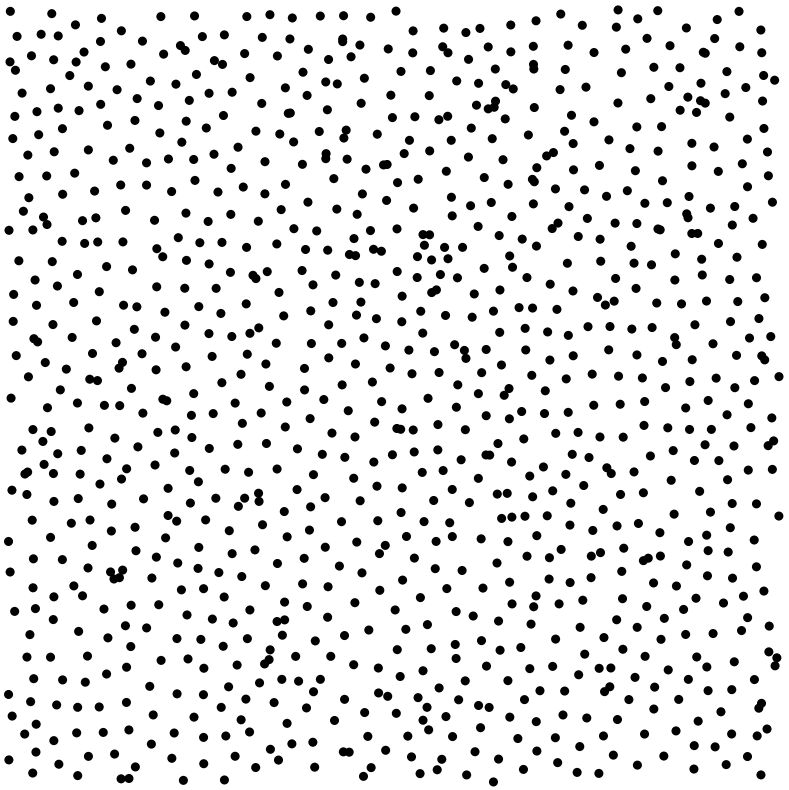
Farthest-Point Optimization

Optimizing Random Seed Points



$\delta_X \approx 0.009,\ \bar{\delta}_X \approx 0.469$

input

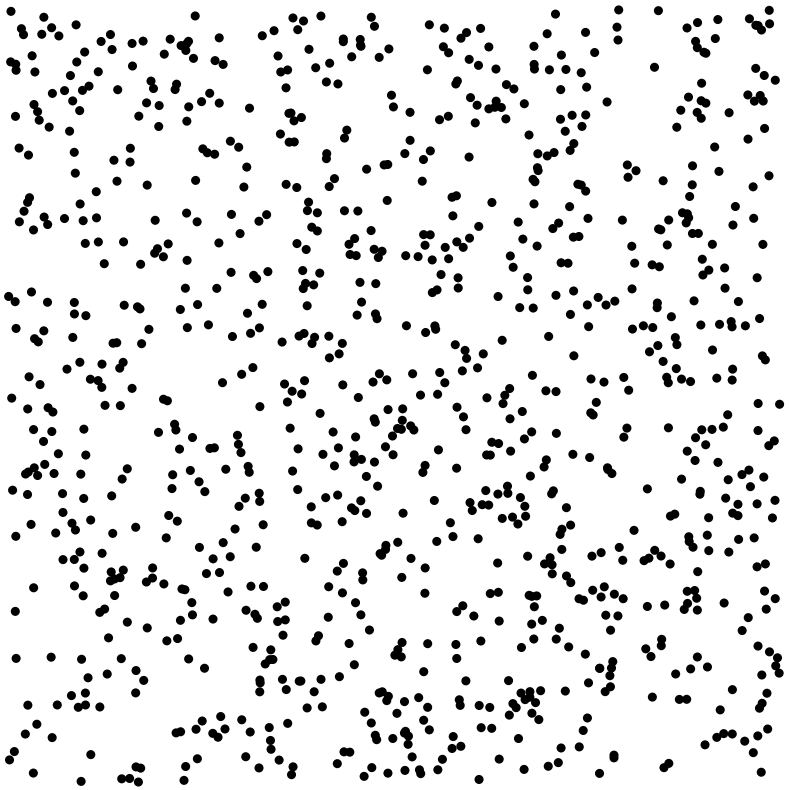


$\delta_X \approx 0.079,\ \bar{\delta}_X \approx 0.756$

1/2 iteration

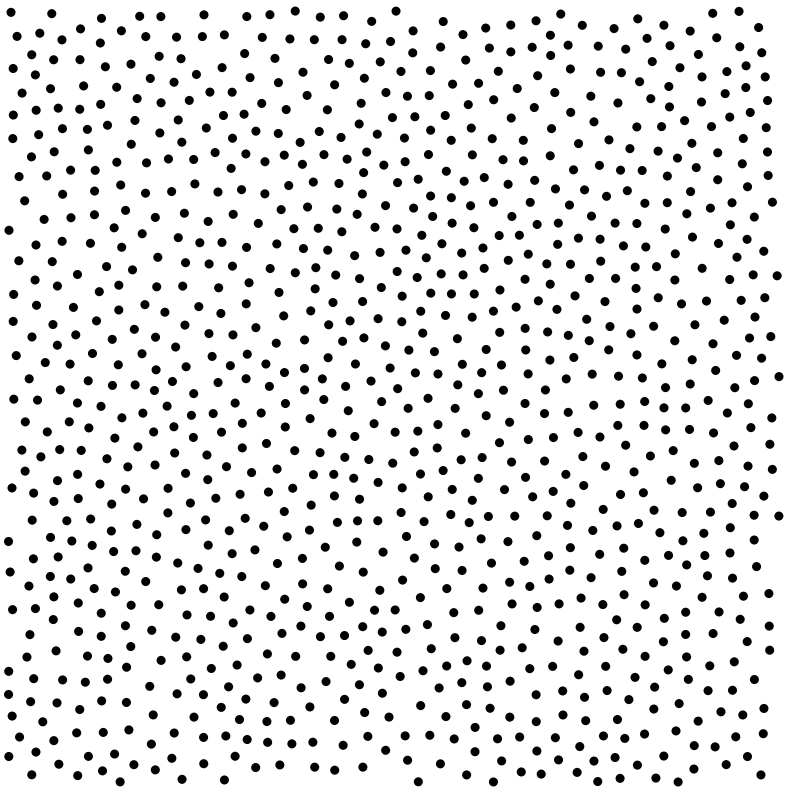
Farthest-Point Optimization

Optimizing Random Seed Points



$\delta_X \approx 0.009,\ \bar{\delta}_X \approx 0.469$

input

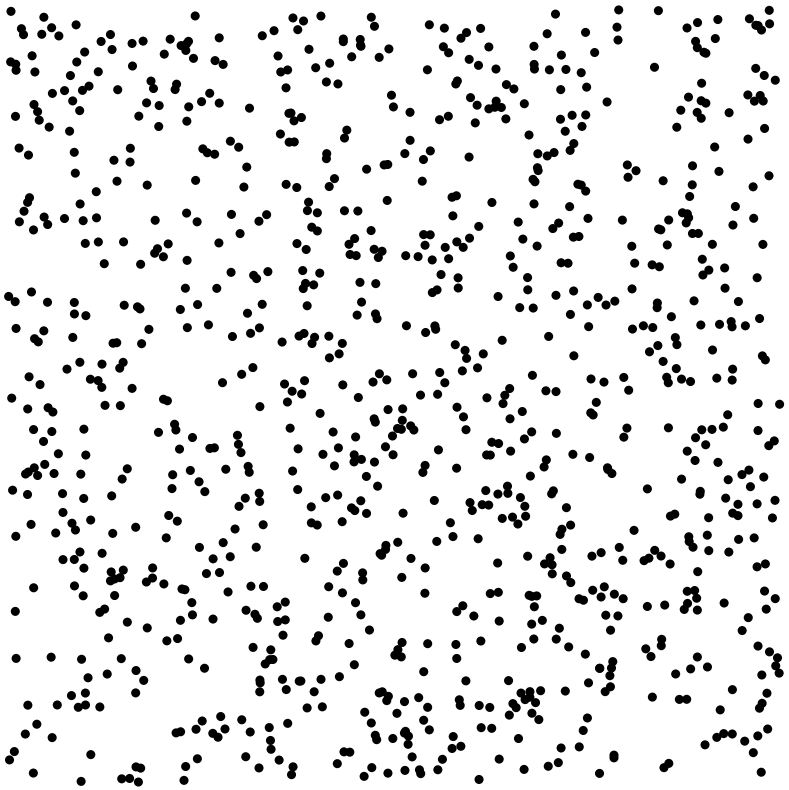


$\delta_X \approx 0.772,\ \bar{\delta}_X \approx 0.865$

1 iteration

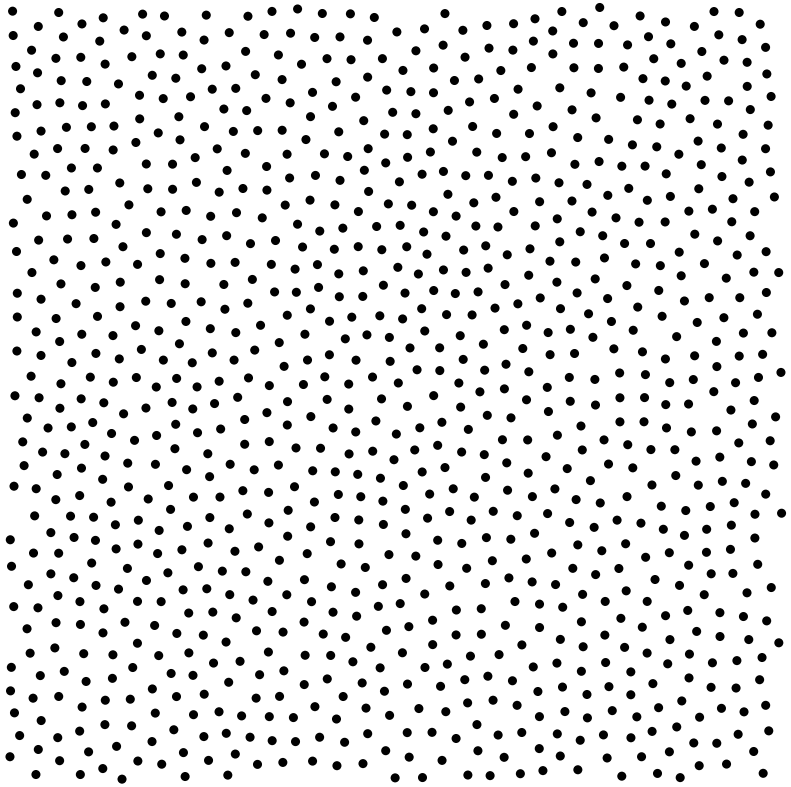
Farthest-Point Optimization

Optimizing Random Seed Points



$\delta_X \approx 0.009,\ \bar{\delta}_X \approx 0.469$

input

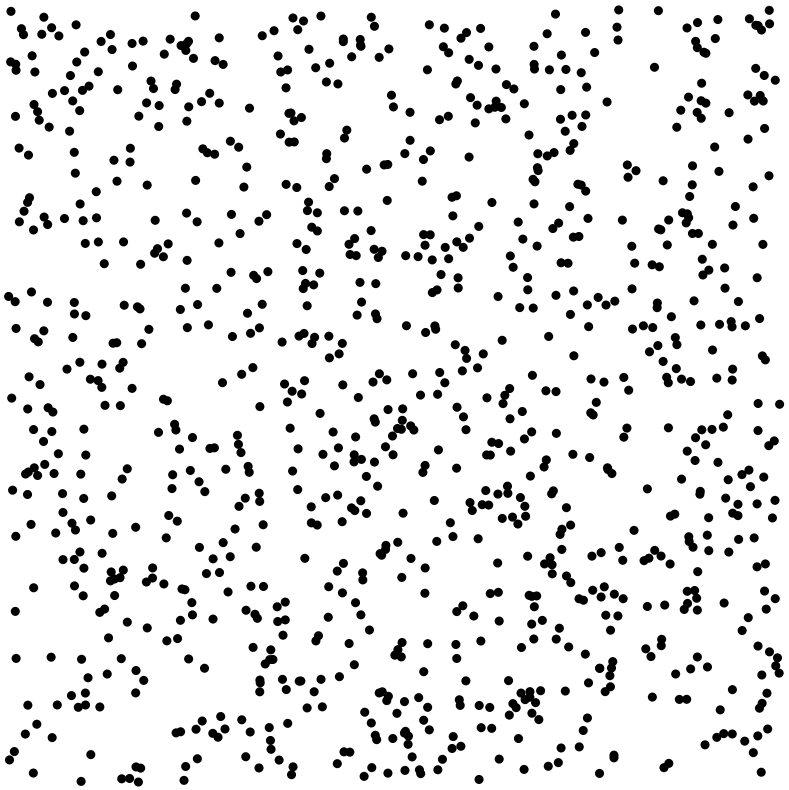


$\delta_X \approx 0.814,\ \bar{\delta}_X \approx 0.905$

2 iterations

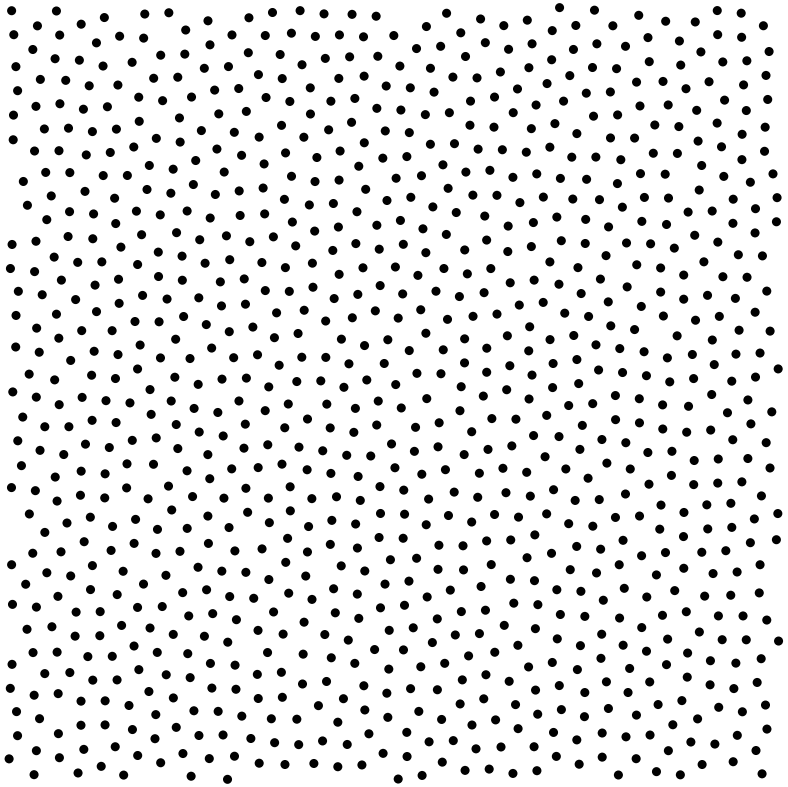
Farthest-Point Optimization

Optimizing Random Seed Points



$\delta_X \approx 0.009,\ \bar{\delta}_X \approx 0.469$

input

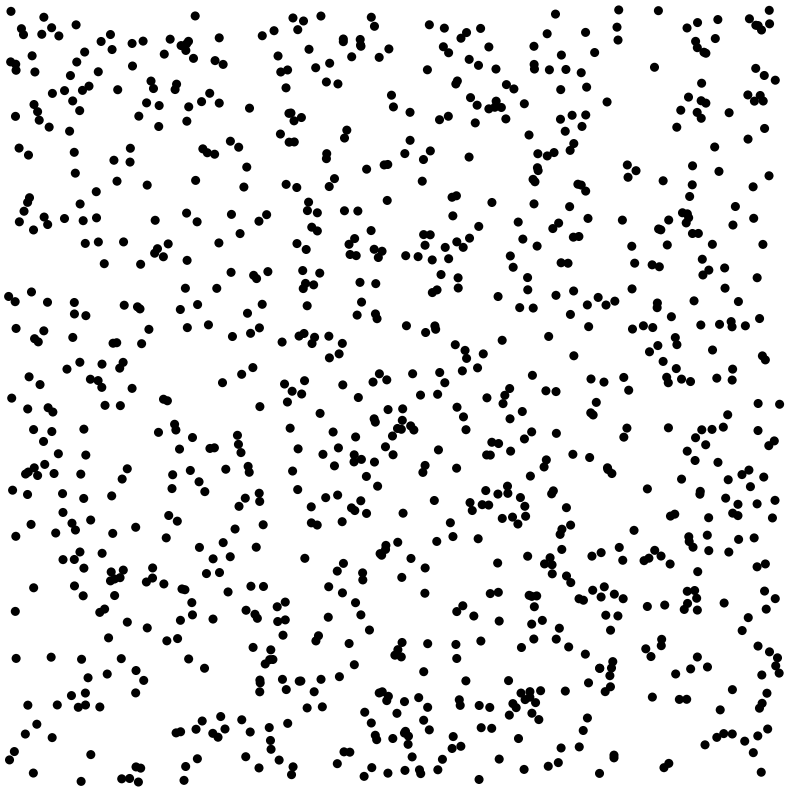


$\delta_X \approx 0.925,\ \bar{\delta}_X \approx 0.932$

63 iterations

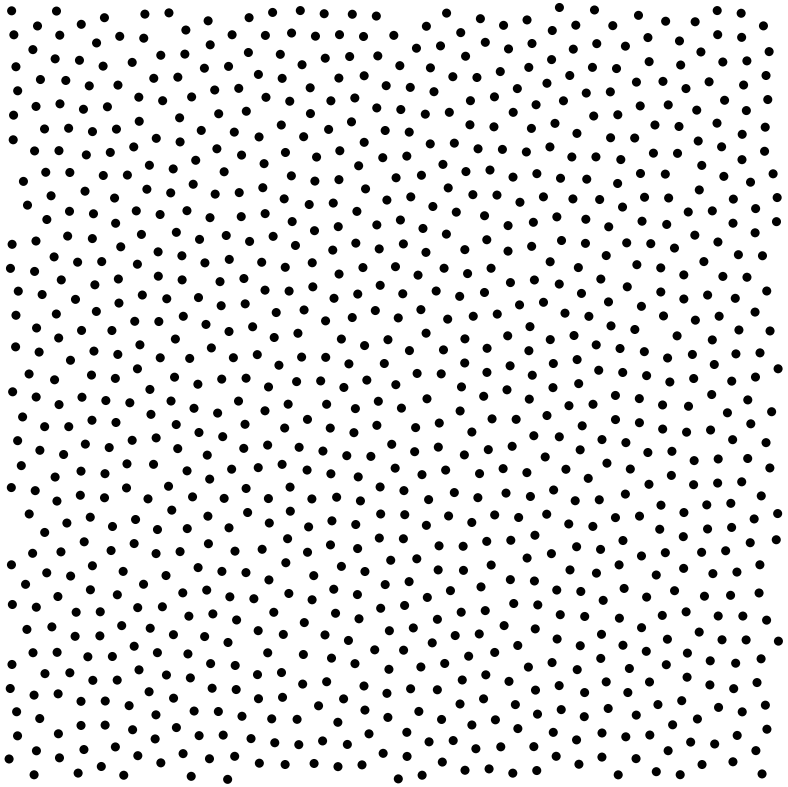
Farthest-Point Optimization

Optimizing Random Seed Points



$\delta_X \approx 0.009,\ \bar{\delta}_X \approx 0.469$

input



$\delta_X \approx 0.930,\ \bar{\delta}_X \approx 0.932$

419 iterations

Farthest-Point Optimization

Convergence

- Moving a point x maximizes by definition its local mindist δ_x
- Since $\bar{\delta}_X \propto \sum \delta_x \;\rightarrow\; \bar{\delta}_X$ increases strictly monotonically
- Stop when $\bar{\delta}_X^{\text{new}} - \bar{\delta}_X^{\text{old}} < \epsilon$ by machine precision

Farthest-Point Optimization

Convergence

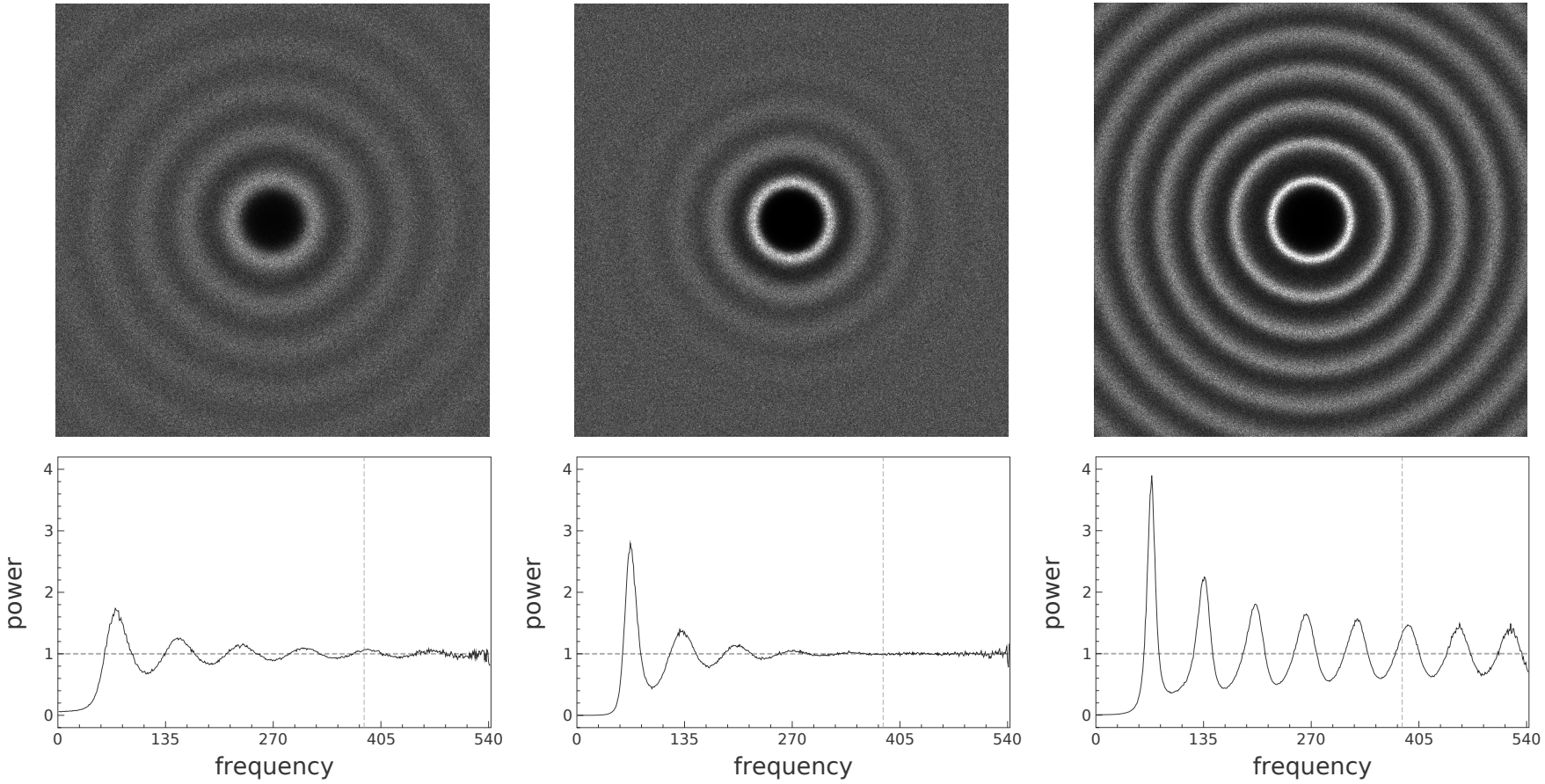
- Moving a point x maximizes by definition its local mindist δ_x
- Since $\bar{\delta}_X \propto \sum \delta_x \;\rightarrow\; \bar{\delta}_X$ increases strictly monotonically
- Stop when $\bar{\delta}_X^{\text{new}} - \bar{\delta}_X^{\text{old}} < \epsilon$ by machine precision

Runtime Complexity

- Dominated by the search operation for the largest circumcircle
- Utilizing a binary tree to track the largest circumcircle yields total complexity of $\mathcal{O}(n \log n)$ for a full iteration
- Delaunay operations in our case typically $\mathcal{O}(1)$ [Erickson 2005]
- In the paper: local variant with $\mathcal{O}(n)$ per iteration but slower convergence

Farthest-Point Optimization

Spectral Analysis



Dart Throwing | [Balzer et al. 2009] | *Farthest-Point Opt.*

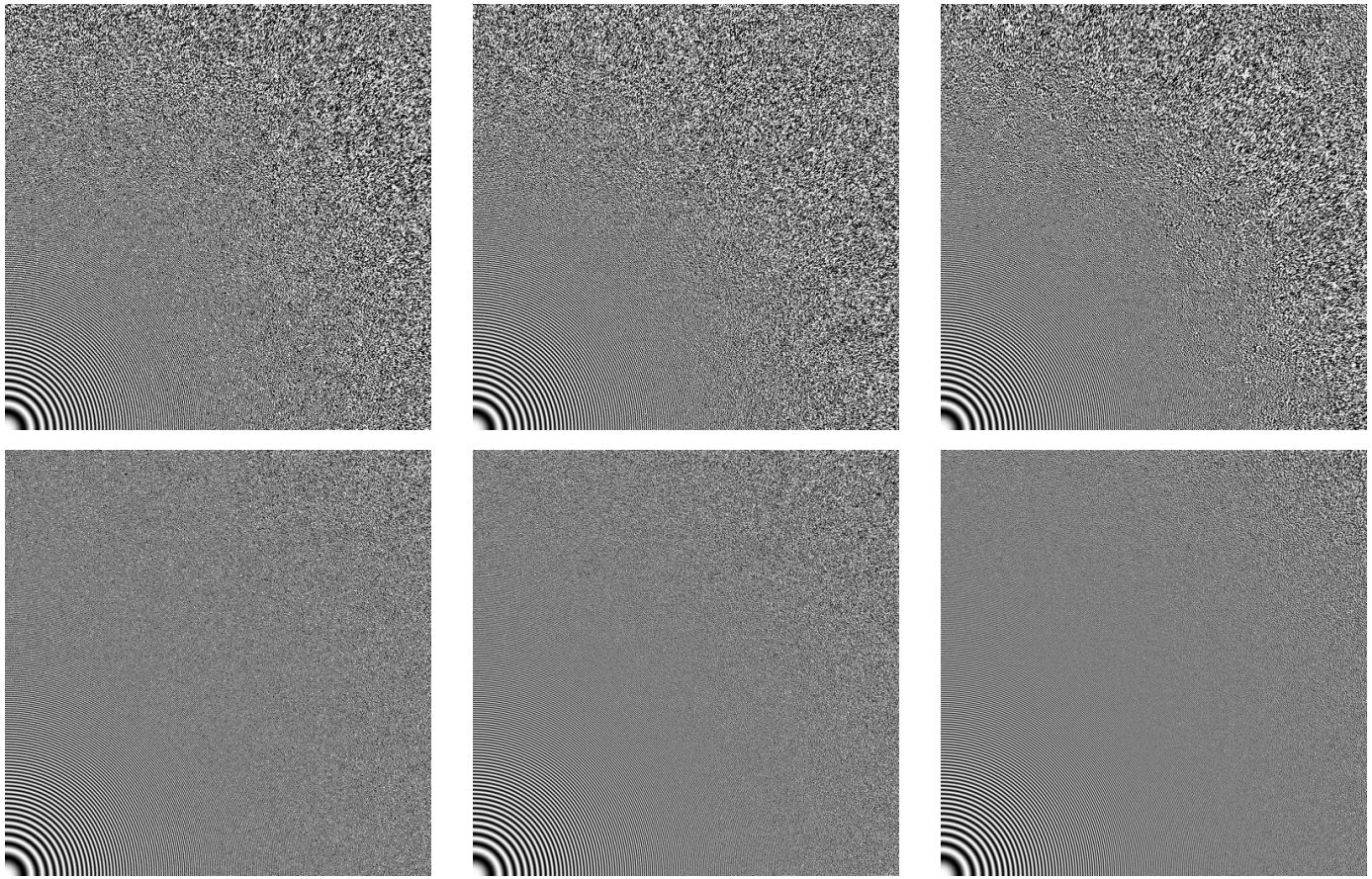
Application

Image Plane Sampling

- High mindist yields high effective Nyquist frequency

≈ 1spp

≈ 2spp



[Dunbar et al. 2006] [Schmaltz et al. 2010] *Farthest-Point Opt.*

Application

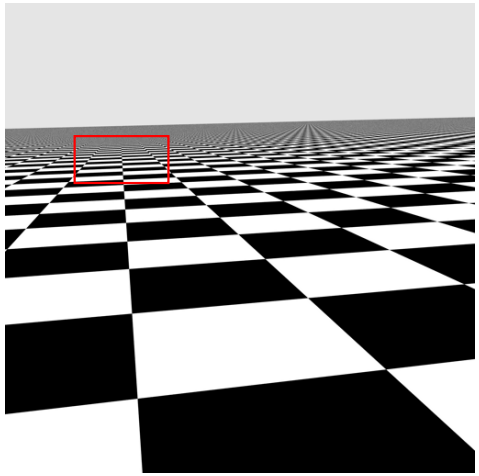
Image Plane Sampling

- FPO random points irregular but uniform
- Good trade-off between noise and coherent aliasing

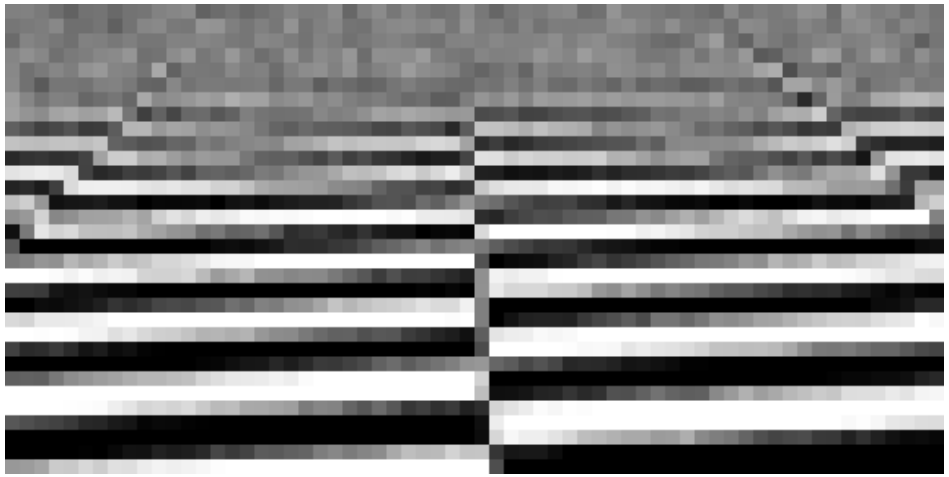
Application

Image Plane Sampling

- FPO random points irregular but uniform
- Good trade-off between noise and coherent aliasing



$\approx$ 16spp



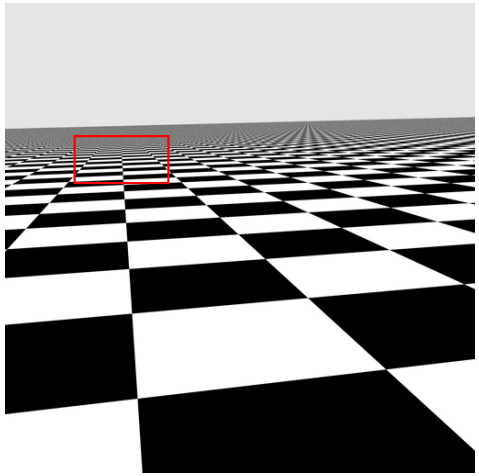
$\mathrm{MSE} = 3.60 \cdot 10^{-3}$

Dart Throwing

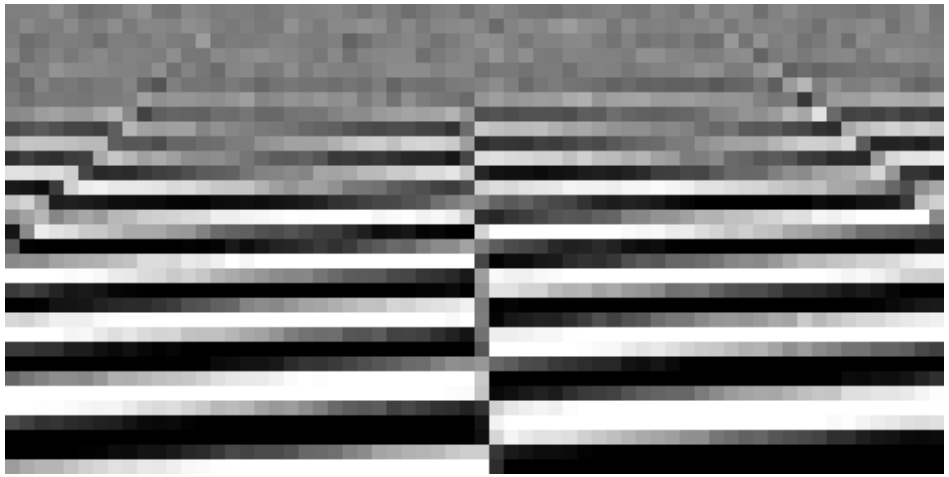
Application

Image Plane Sampling

- FPO random points irregular but uniform
- Good trade-off between noise and coherent aliasing



$\approx$ 16spp



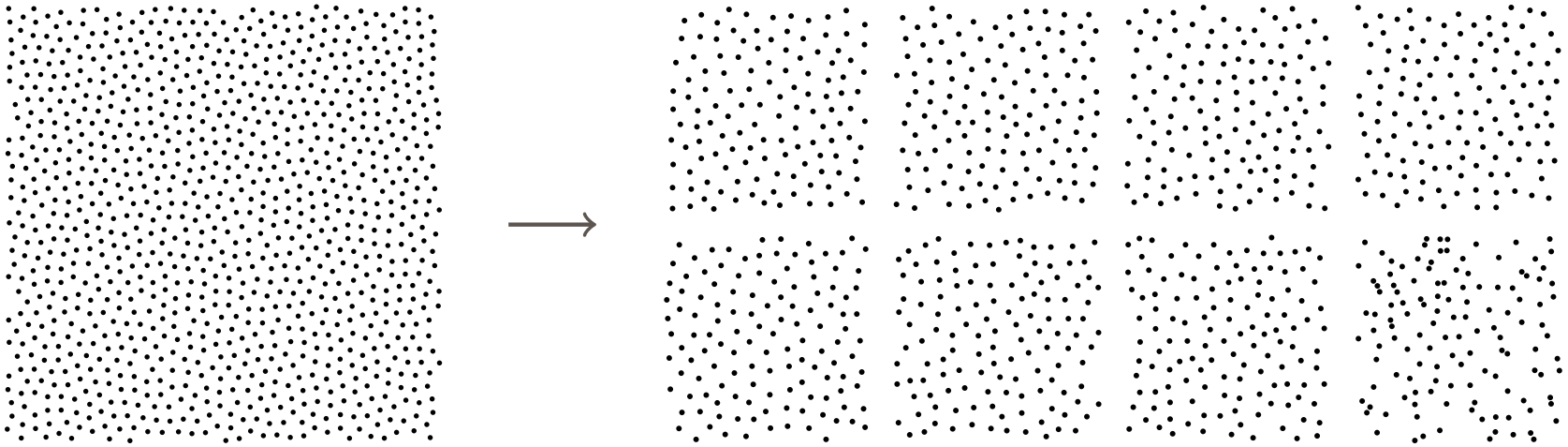
$\mathrm{MSE} = 2.58 \cdot 10^{-3}$

Farthest-Point Optimized

Extension

Extension

Partition Algorithm

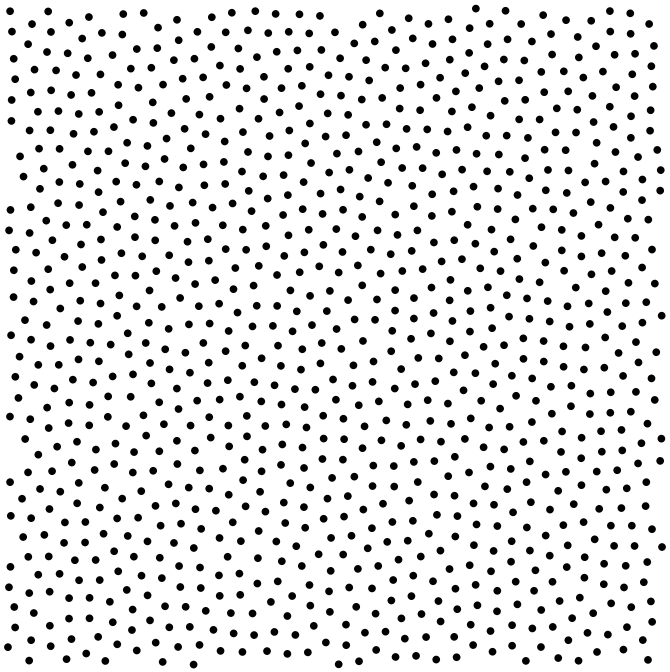


1024 points $\longrightarrow$ 8×128 points

- Want each subset as well as their union well-distributed
- Idea: partition input set into optimized subsets of equal size

Extension

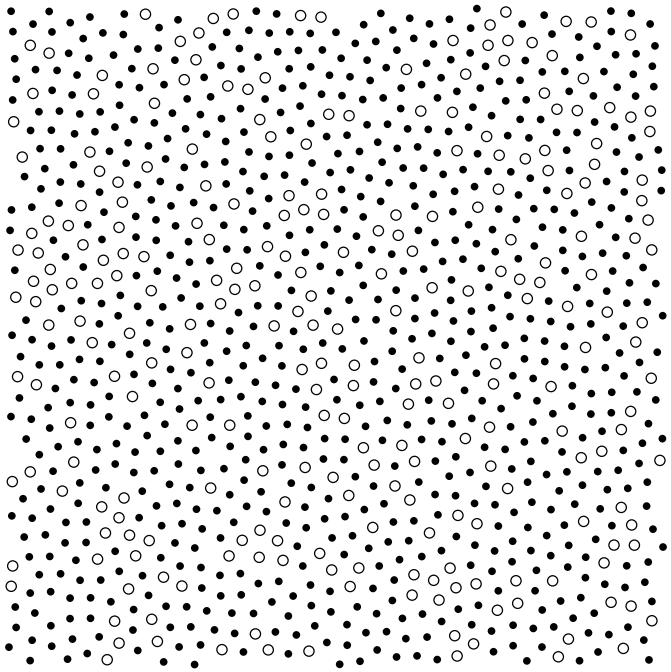
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

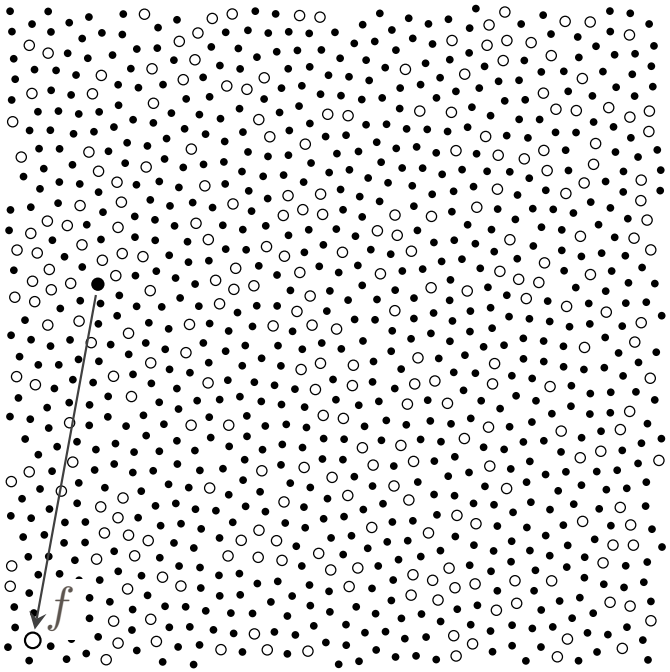
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

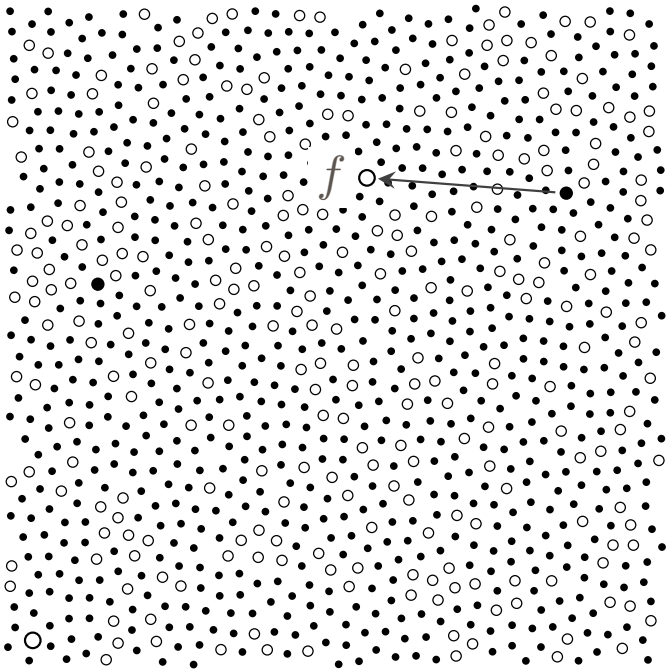
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

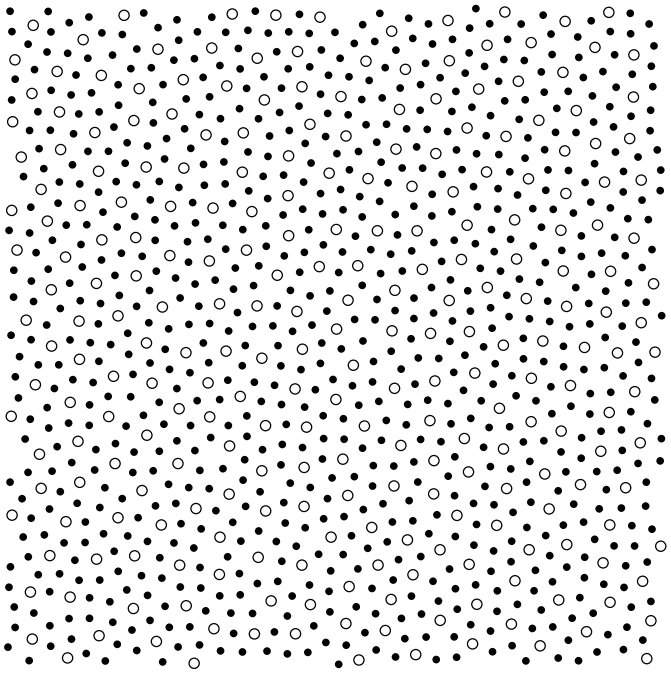
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

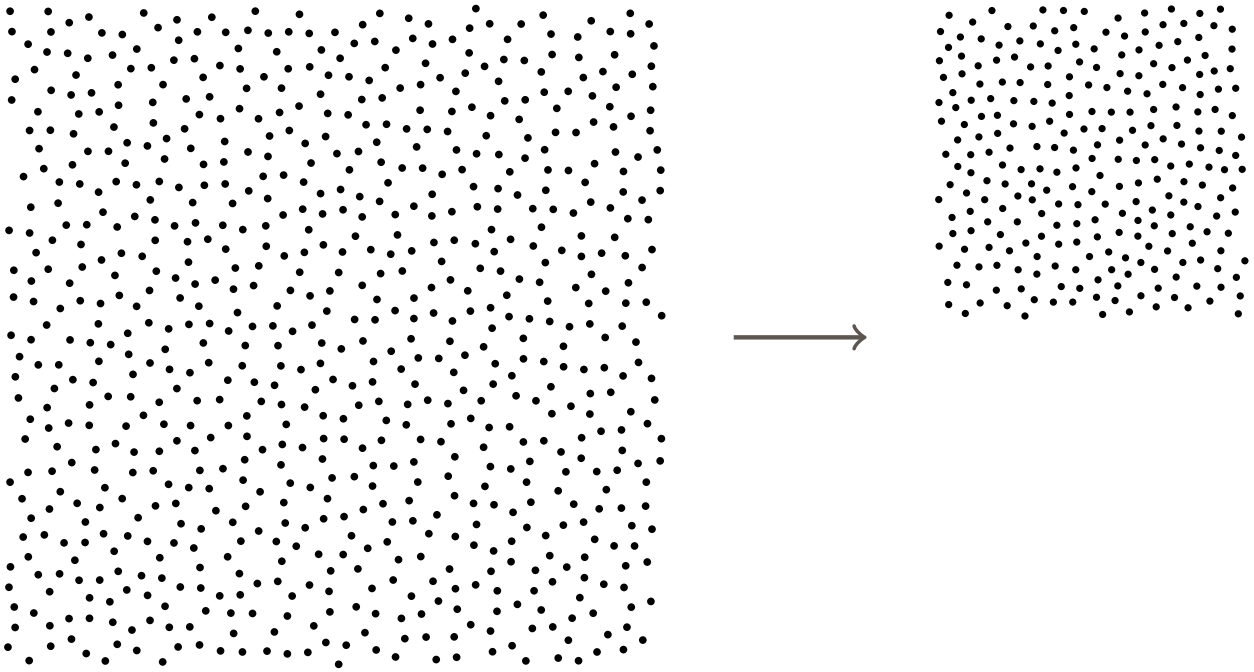
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

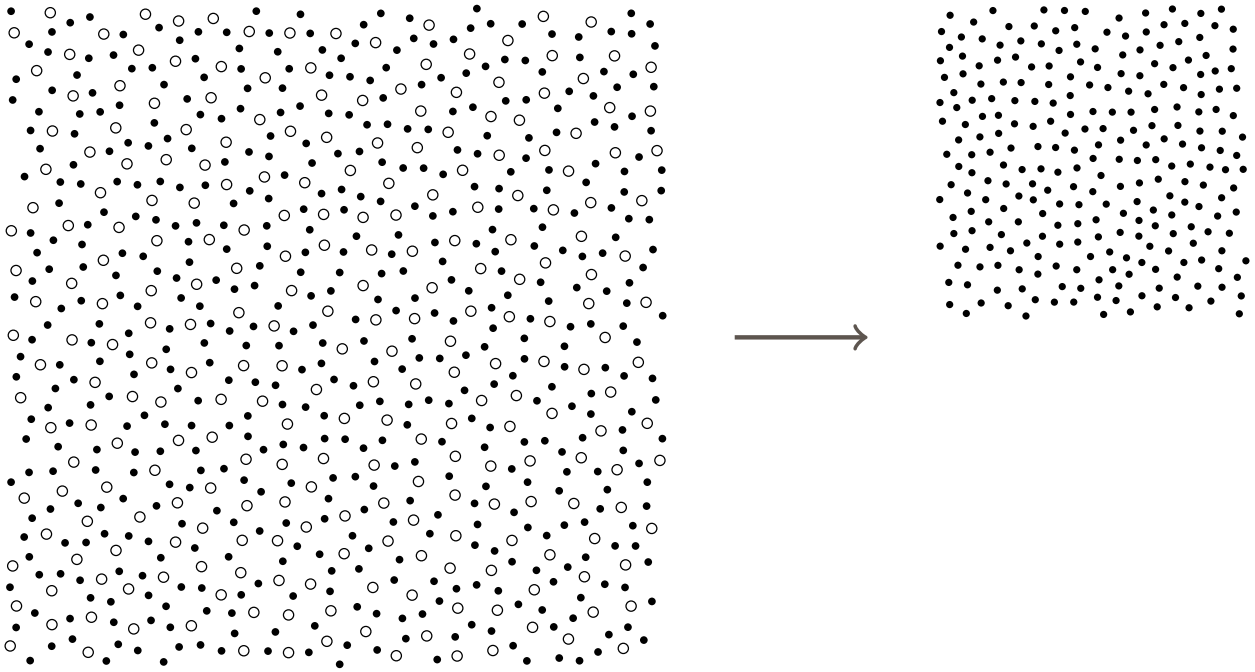
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

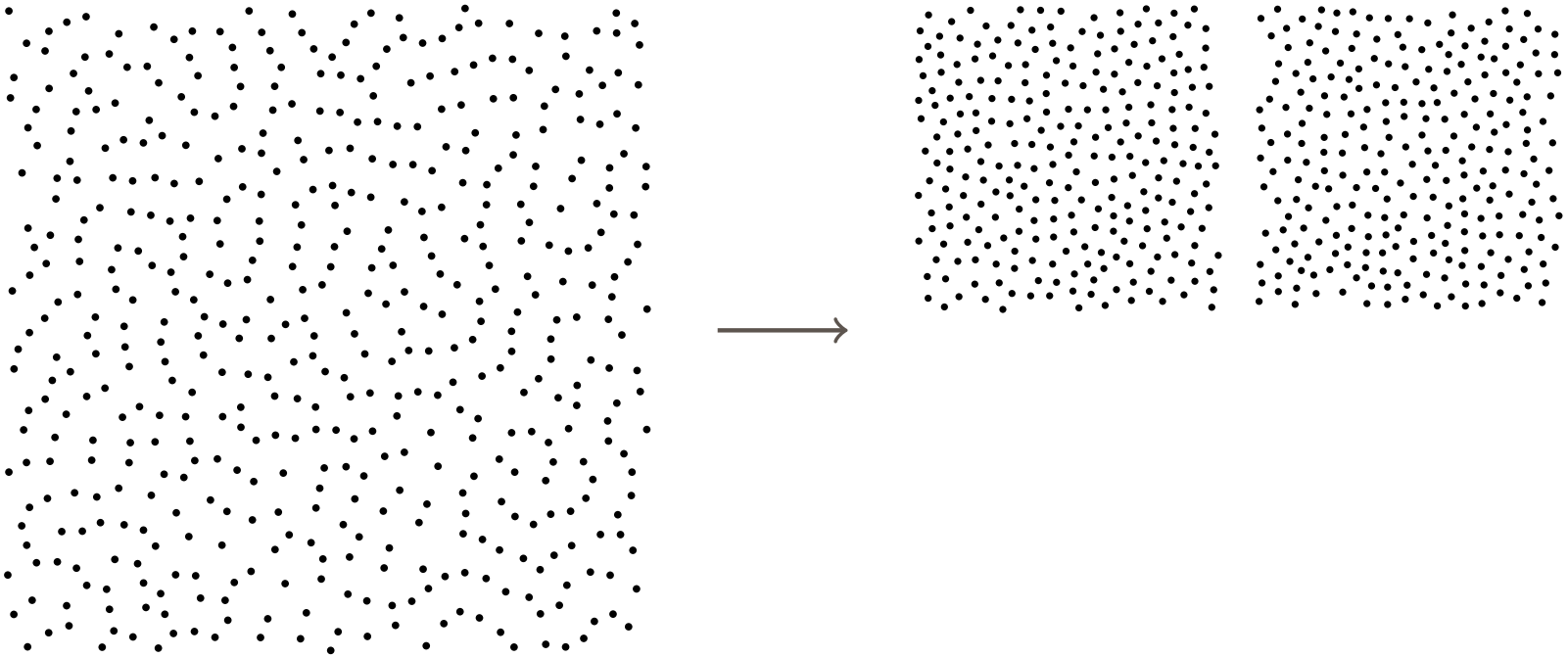
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

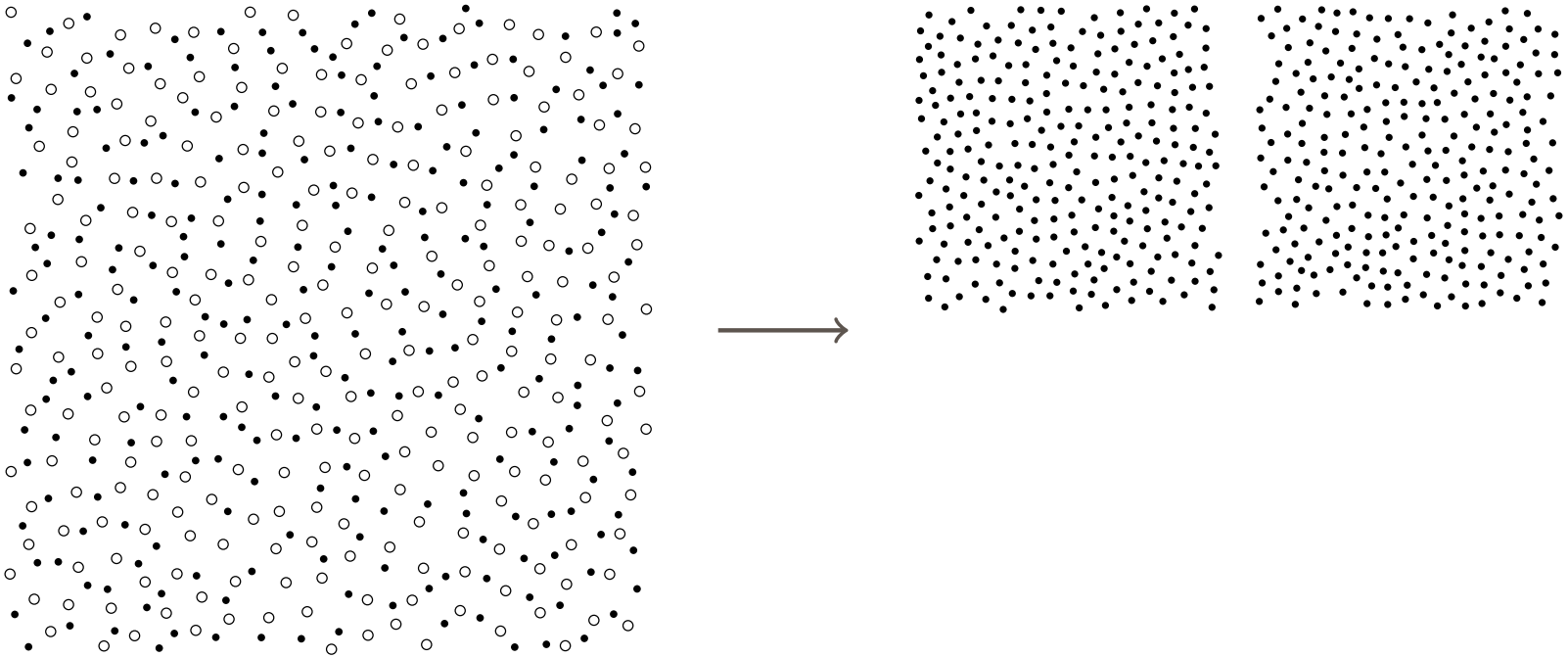
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

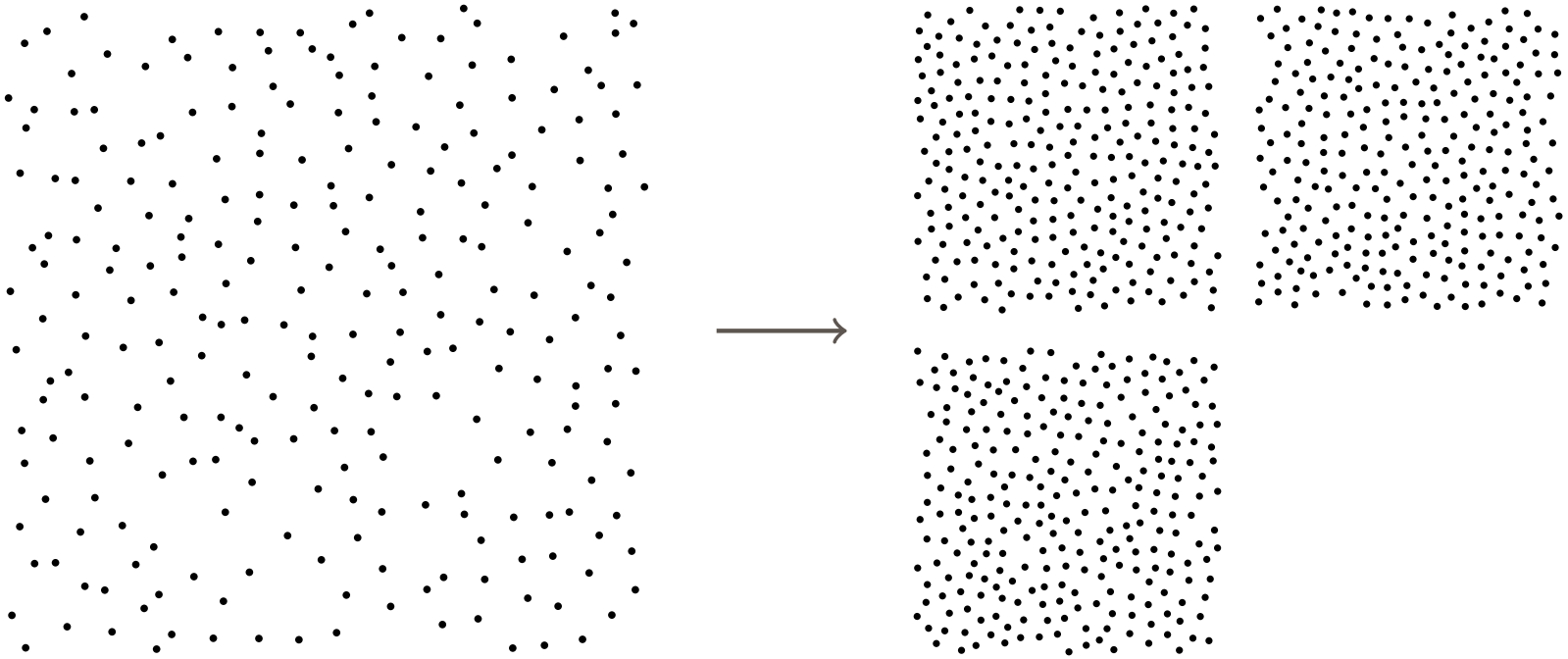
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

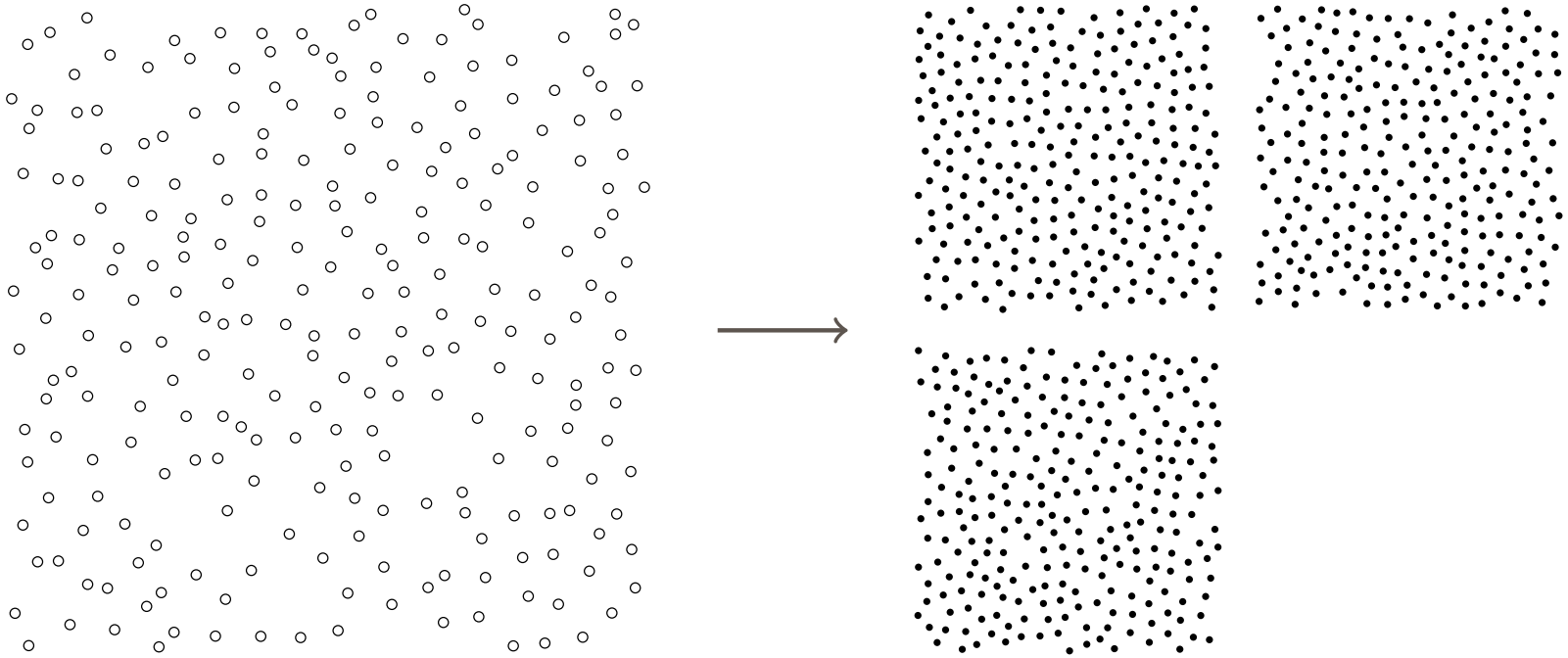
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

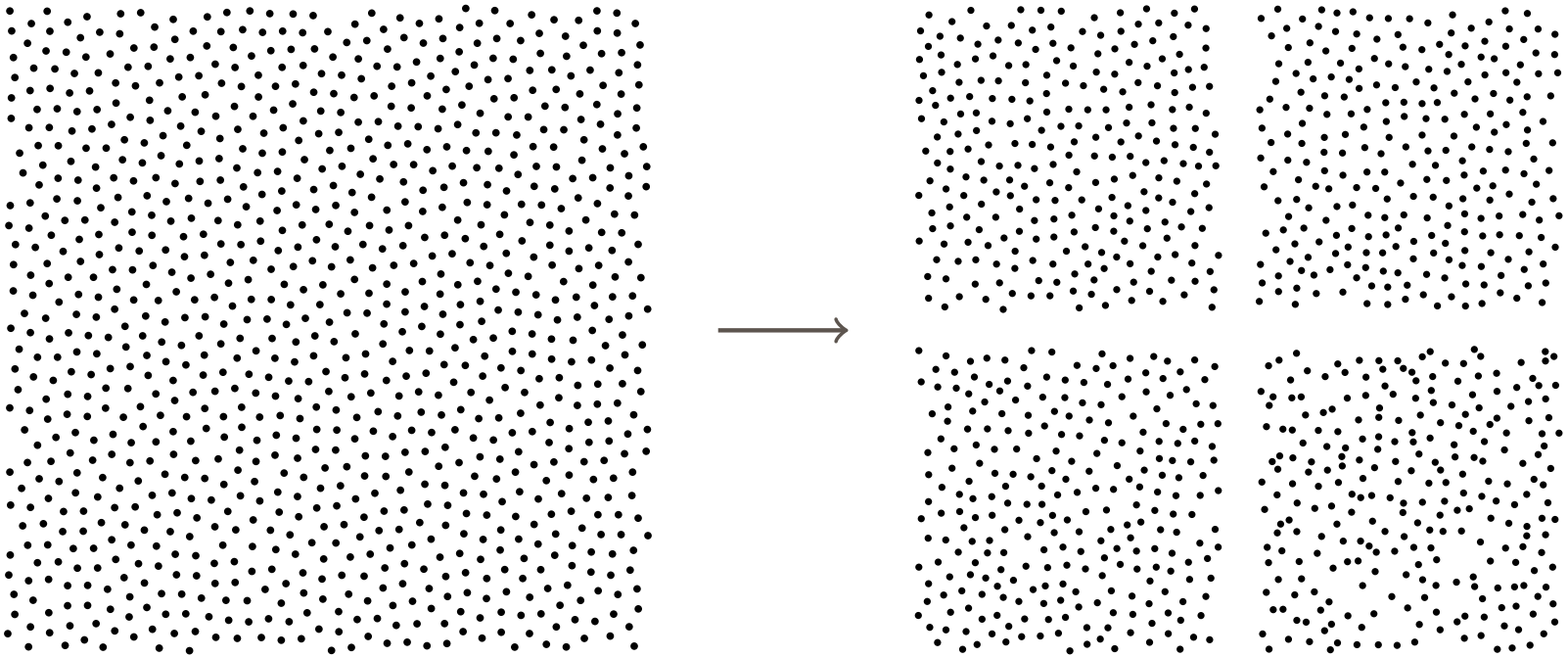
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

Extension

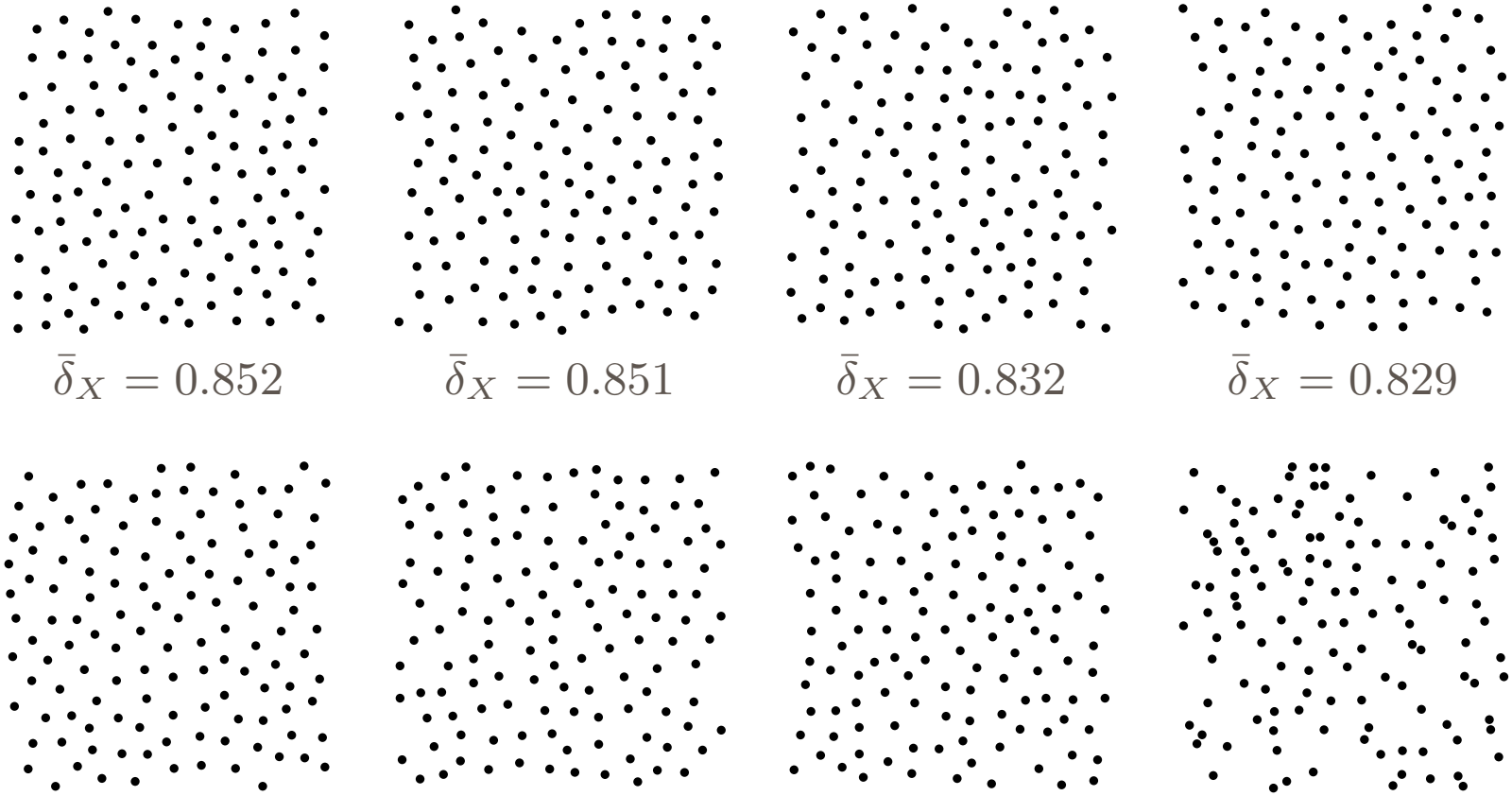
Partition Algorithm



- For each subset: discrete space variant of main FPO algorithm
- Main difference: f now bound to a point $f \in X$
- Greedy approach: construct subsets sequentially

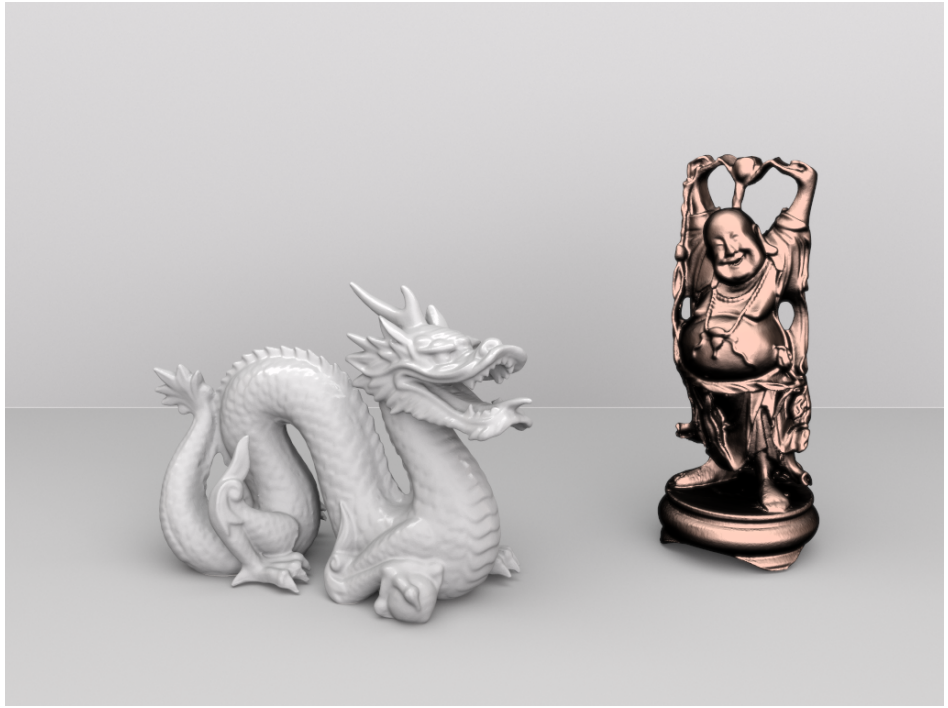
Extension

Partition Algorithm



Application

Trajectory Splitting

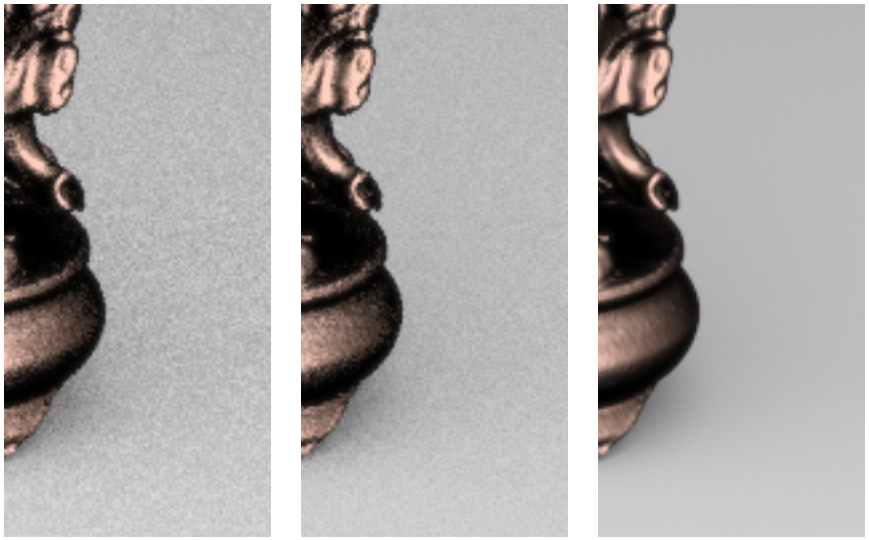


- Direct light estimation by trajectory splitting
- Precompute FPO points and necessary partitions
- Unbiased estimators by random shifts on unit torus

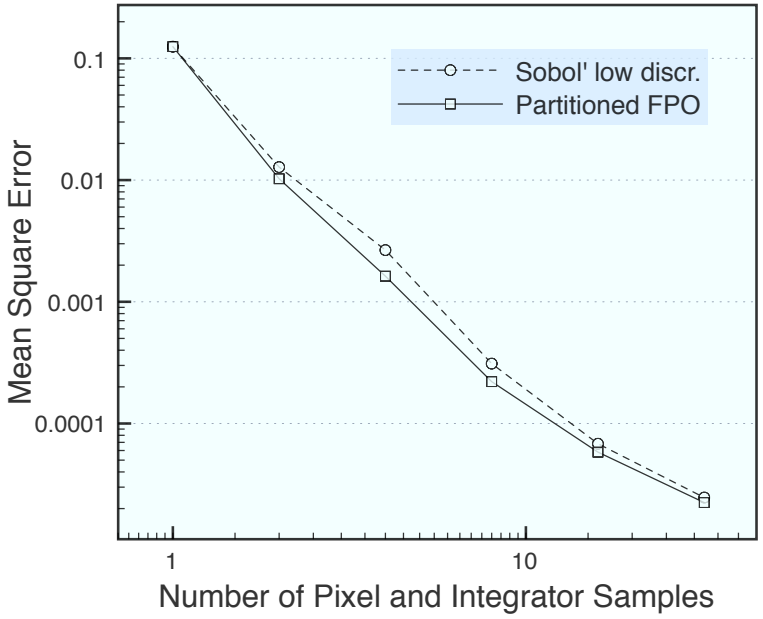
Application

Trajectory Splitting

4 × 4 samples

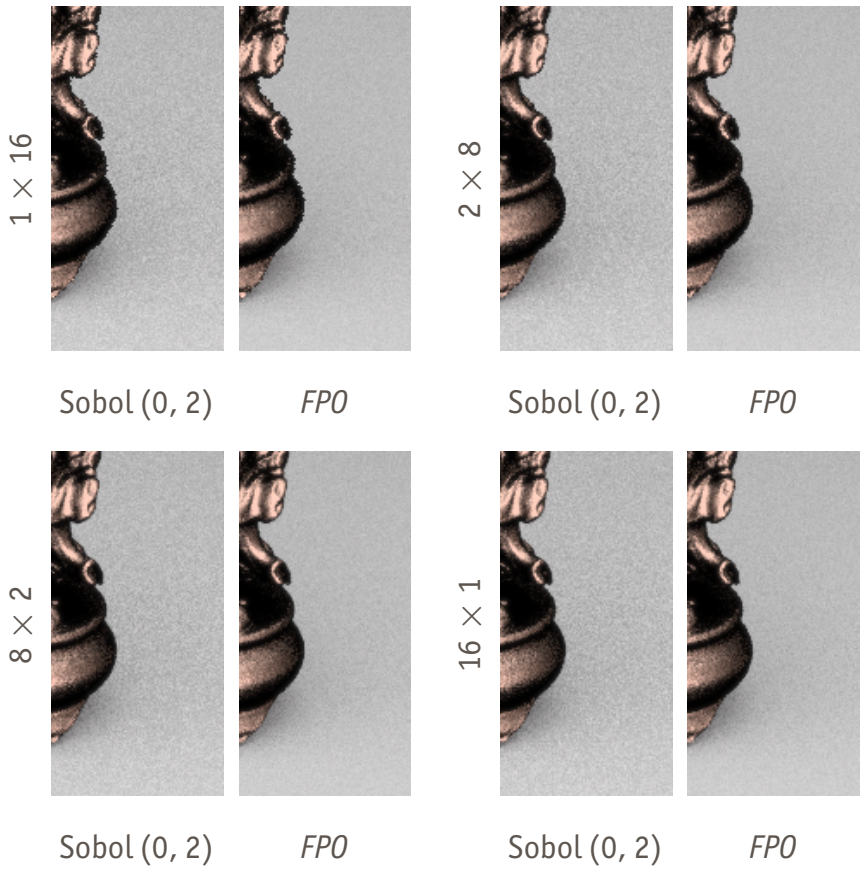


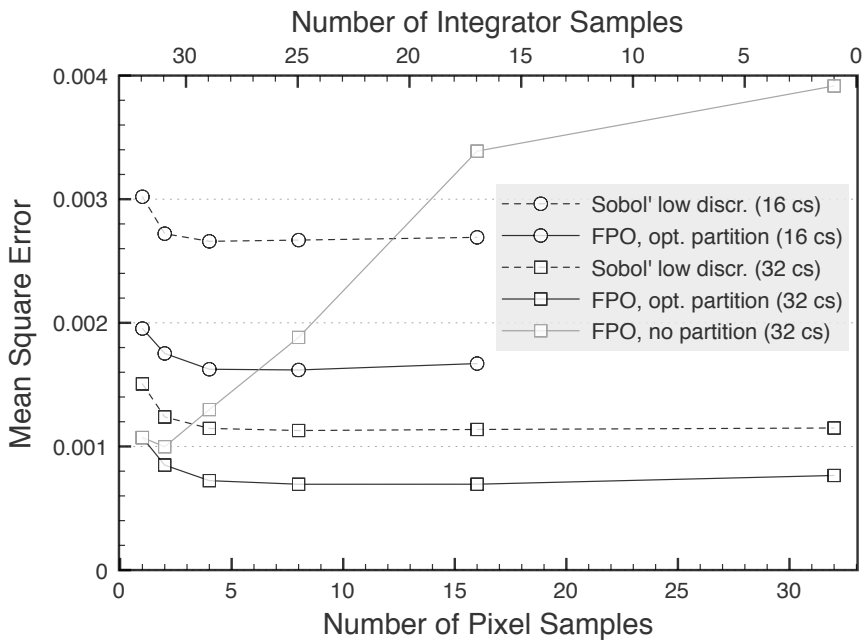
Sobol (0, 2) *FPO* reference



Application

Trajectory Splitting



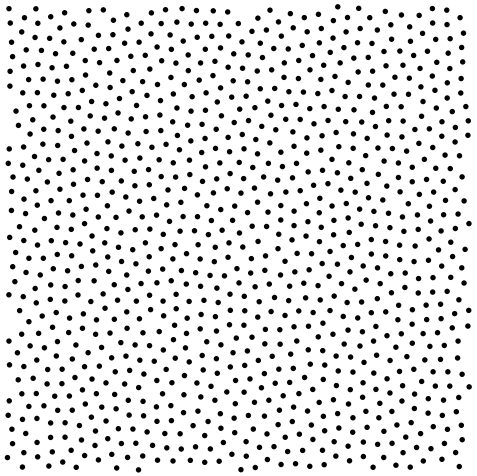


Conclusion

Conclusion

Farthest-Point Optimized Point Sets

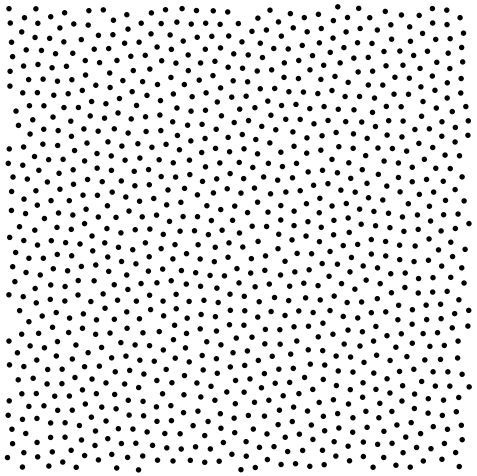
- Uniform but irregular
- Higher mindist than previous methods
- Good trade-off between noise and coherent aliasing
- Optimal blue noise?



Conclusion

Farthest-Point Optimized Point Sets

- Uniform but irregular
- Higher mindist than previous methods
- Good trade-off between noise and coherent aliasing
- Optimal blue noise?



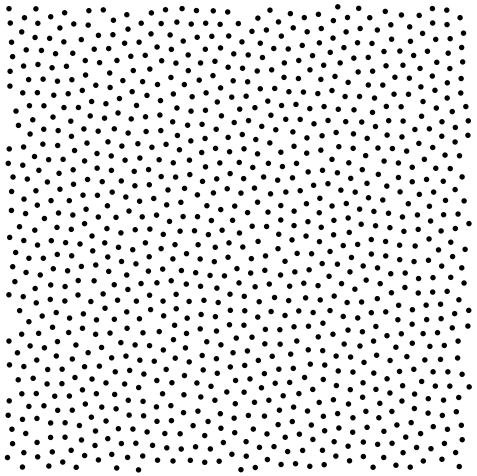
Partition Algorithm

- Good results but greedy approach leaves room for improvement

Conclusion

Farthest-Point Optimized Point Sets

- Uniform but irregular
- Higher mindist than previous methods
- Good trade-off between noise and coherent aliasing
- Optimal blue noise?



Partition Algorithm

- Good results but greedy approach leaves room for improvement

Future Work

- Points on bounded surfaces or triangulated domains
- Behavior in higher dimensions or for non-Euclidean metrics

Thank you for your attention.

References

- Balzer, M., Schlömer, T., and Deussen O., 2009.
  Capacity-Constrained Point Distributions: A Variant of Lloyd's Method.
  ACM Trans. Graph. (Proceedings of SIGGRAPH 2009) 28, 3, 86:1–8.
- Cook, R. L., 1986.
  Stochastic Sampling in Computer Graphics.
  Computer Graphics (Proc. of SIGGRAPH 86) 5, 1, 51–72.
- Du., Q., Faber, V., and Gunzburger, M., 1999.
  Centroidal Voronoi tessellations: Applications and Algorithms.
  SIAM Review 41, 4, 637–676.
- Dunbar, D., and Humphreys, G., 2006.
  A Spatial Data Structure for fast Poisson-Disk Sample Generation.
  ACM Trans. Graph. (Proceedings of SIGGRAPH 2006) 25, 3, 503–508.

References

- Eldar, Y., Lindenbaum, M., Porat, M., and Zeevi, Y. Y., 1997.
  The Farthest Point Strategy for Progressive Image Sampling.
  IEEE Trans. Image Process. 6, 9, 1305–1315.
- Erickson, Jeff, 2005.
  Dense Point Sets Have Sparse Delaunay Triangulations.
  Discrete Comput. Geom. 1, 33, 83–115.
- Grünschloß, L., and Keller, A., 2009.
  (t, m, s)*-nets and maximized minimum distance, part II*.
  Monte Carlo and quasi-Monte Carlo Methods 2008, 395–409.
- Lagae, A., and Dutré, P., 2008.
  A Comparison of Methods for Generating Poisson-Disk Distributions.
  Computer Graphics Forum 27, 1, 114–129.

References

- Lloyd, S. P., 1982.
  Least Square Quantization in PCM.
  IEEE Transactions on Information Theory, 28, 2, 129–137.
- Mitchell, D. P., 1991.
  Spectrally Optimal Sampling for Distribution Ray Tracing.
  Computer Graphics (Proc. of SIGGRAPH 91), 157–164.
- Schmaltz, C., Gwosdek, P., Bruhn, A., and Weickert, J., 2010.
  Electrostatic Halftoning.
  Computer Graphics Forum 29, 8, 2313–2327.