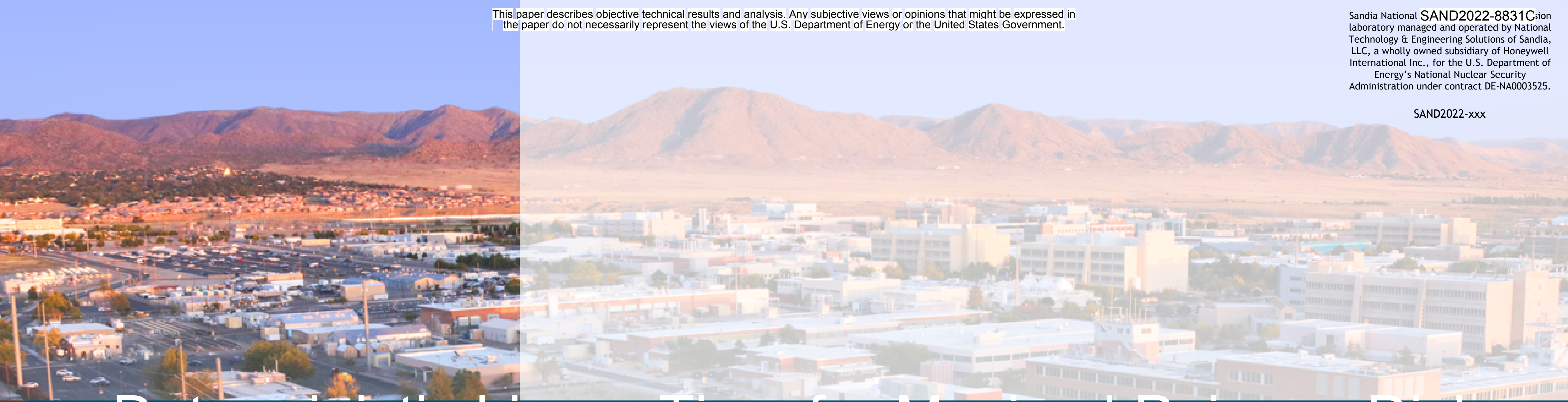
Sandia National SAND2022-8831C sion laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2022-xxx



Deterministic Linear Time for Maximal Poisson-Disk Sampling using Chocks without Rejection or Approximation

SGP 6 July 2022

20 minutes

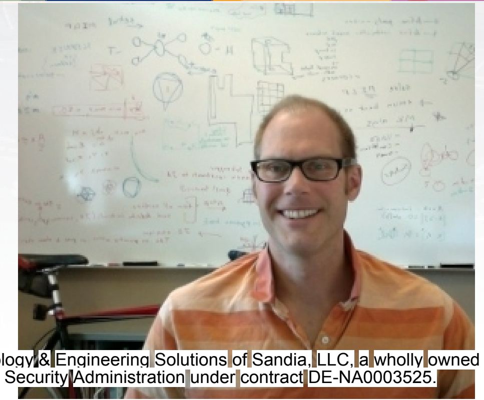
CCR Center for Computing Research

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY | Office of S

Scott A. Mitchell

sandia.gov/~samitch



Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Deterministic Linear Time for Maximal Poisson-Disk Sampling using Chocks without Rejection c

What's new?

- Chock geometry

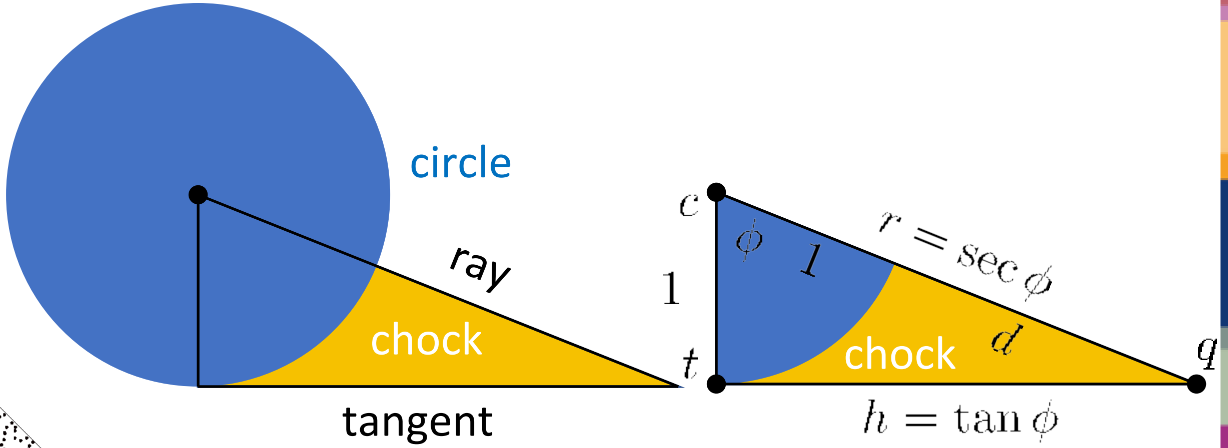
  Area $A(\phi) = (\tan\phi - \phi)/2$

  $A^{-1}(uA) =$ 5 Newton's iterations

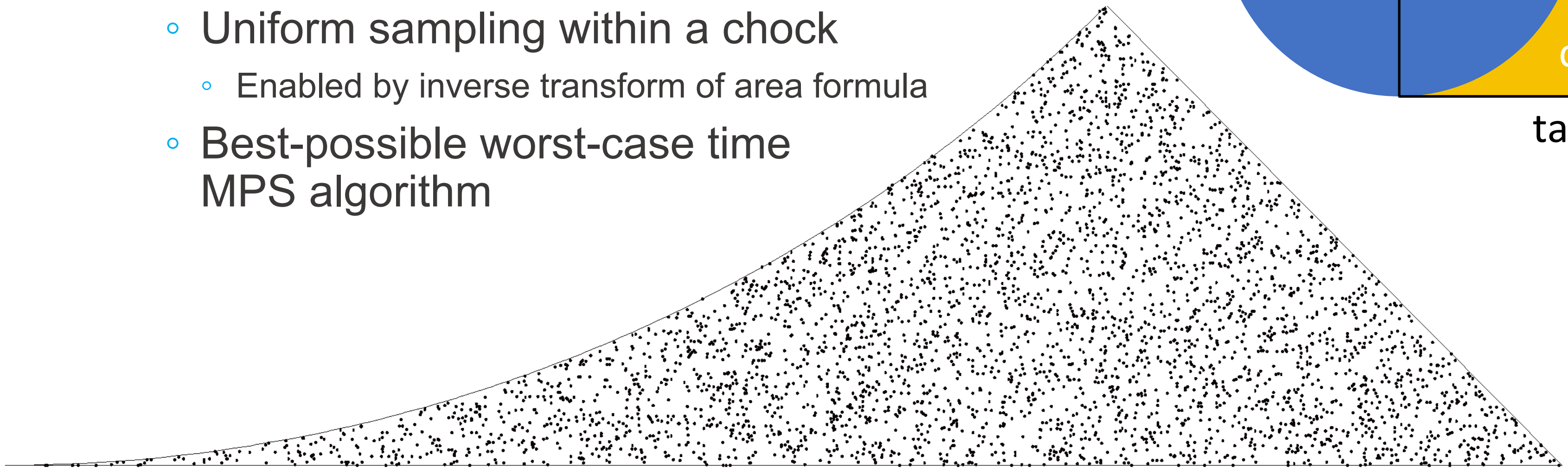
  $$\phi_0 = \sqrt[3]{6uA}$$

  $$\phi_{i+1} = \phi_i - \frac{\tan\phi_i - \phi_i - 2uA}{\tan^2\phi_i}$$

  $$r_s = \sqrt{v\tan^2\phi_s + 1}$$



- Uniform sampling within a chock
  - Enabled by inverse transform of area formula
- Best-possible worst-case time MPS algorithm



Poisson-disk Sampling $\Longrightarrow$ Dart Throwing Algorithm

time of arrival order of arrival

Poisson: points arrive at constant mean rate

- Defined by process
- Insert random disks, build set X
- Disks don't contain another sample
- Poisson: points arrive at constant mean rate
  - Dart throwing equivalent output: unbiased order or arrival

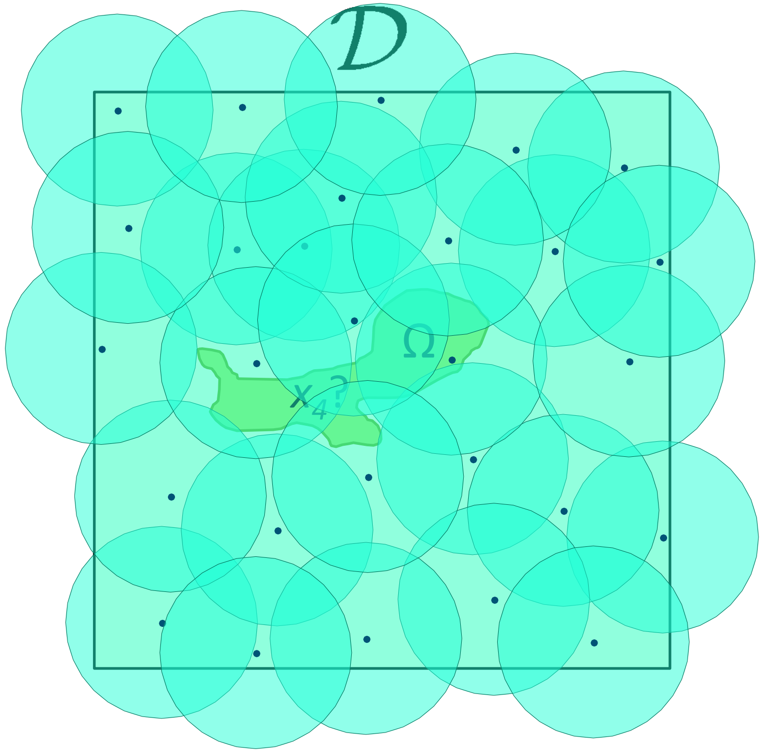
Empty disk: $\forall x_i, x_j \in X, x_i \neq x_j : ||x_i - x_j|| \geq r$

$\forall \Omega \in \mathcal{D}$:

Poisson-process time: $PDF(t;\Omega) = \text{Area}(\Omega)e^{-t\text{Area}(\Omega)}$

Dart order of arrival: $P(x_i \in \Omega) = \text{Area}(\Omega)/\text{Area}(D)$

Maximal: $\forall x \in \mathcal{D}, \exists x_i \in X : ||x - x_i|| < r$



Definition or Process?

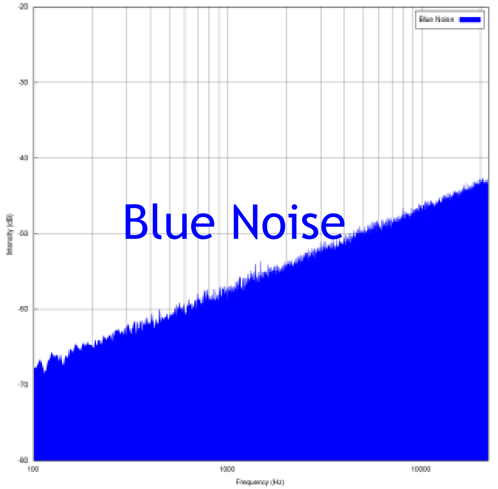
Which would you rather have?

Definition of desired output
- Sorted order
  - So you can discover quicksort

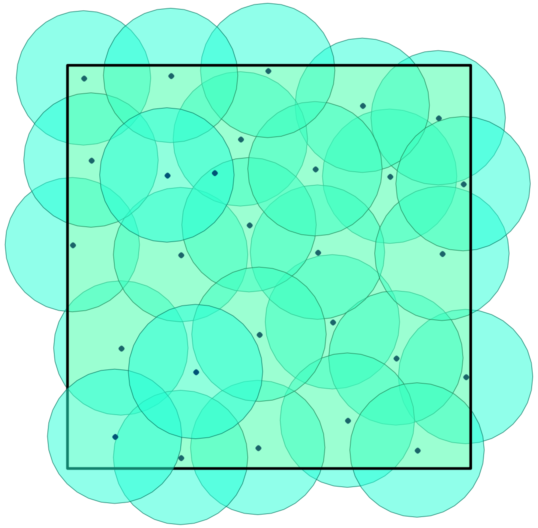
Process to obtain it (e.g. algorithm)
- Bubblesort algorithm O(n^2)

Another example:
- Ax=b
- Gaussian Elimination O(n^3)

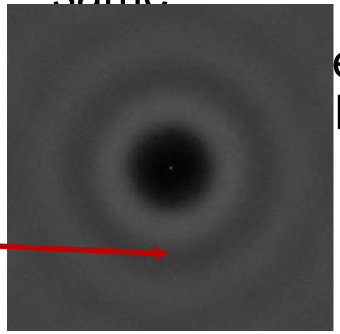
Blue-noise **is a desired output**
Fourier transform of a distribution
- **Def.** power density
  increases 3.01 dB per octave
  (density proportional to *f*)
  over a finite frequency range.
- In computer graphics,
  "blue noise" is **loosely any noise**
  with minimal low frequency components
  and
  no concentrated spikes in energy.
  [Wikipedia]



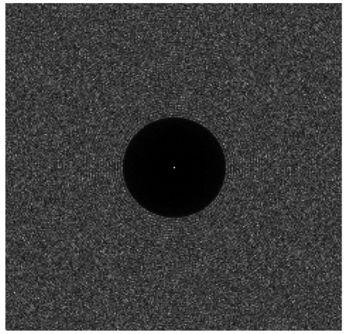
Poisson-disk Sampling **is a process** for generating a distribution
- Output resembles blue-noise, but is not same
  efinitio
  but it lo



Oscillating rings



Poisson-disk
[DH06a]



Blue Noise
[HSD13]

Poisson-disk Sampling vs. Blue Noise

Graphics community history

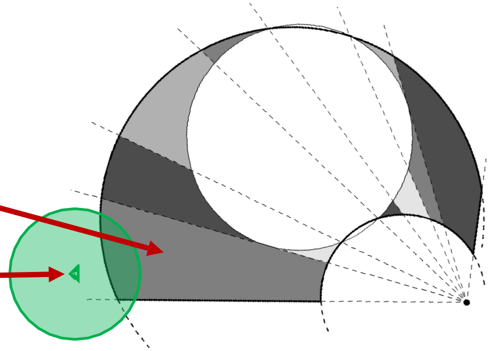
- Early claim that distribution of human retina receptors was Poisson-disk
- When looking at images (e.g. stippling), Blue-noise
  - Randomness avoids aliasing artifacts
  - Separation provides efficiency, no redundant samples
- Dart throwing process produces the same points as the Poisson-disk process
    - Simple algorithm
    - Output close-enough to blue-noise
  - **Not** saturation in finite runtime.
    - Fix: only generate samples from the uncovered subdomain

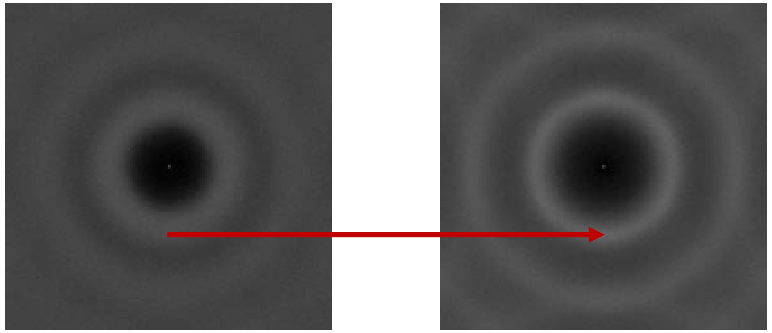
Scalloped Sectors

ScallopMDS, “Using Scalloped Sectors to Generate Poisson-disk Sampling Patterns” SIGGRAPH 2006.

Advancing front

- Next disk distance [r, 2r] from a prior disk
- Never here,
  so not Poisson-disk process





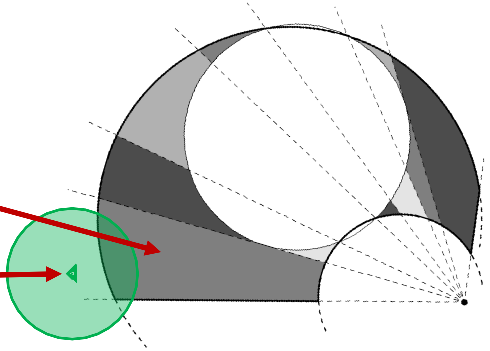
Brighter rings
Figures from their paper

Poisson-disk

Scallop-disk
not Poisson-disk output

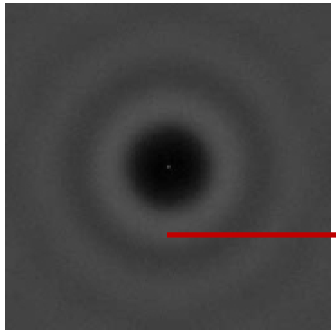
Scalloped Sectors

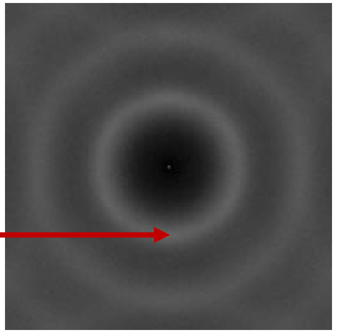
ScallopMDS, “Using Scalloped Sectors to Generate ~~Poisson~~-disk Sampling Patterns” SIGGRAPH 2006.



Advancing front

- Next disk distance [r, 2r] from a prior disk
- Never here,
  so not Poisson-disk process



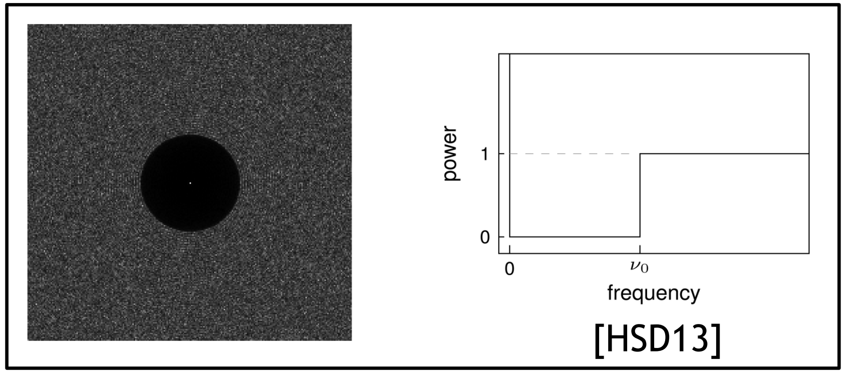


Brighter rings

Figures from their paper

Poisson-disk

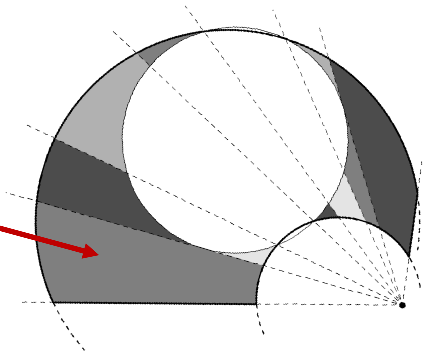
Scallop-disk
not Poisson-disk output



“Step Blue Noise” = real blue noise

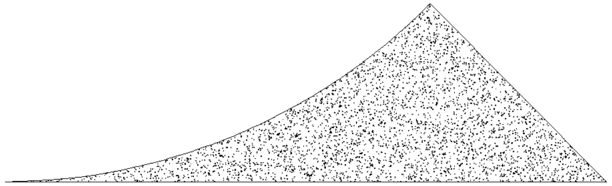
Scalloped Sectors

ScallopMDS, “Using Scalloped Sectors to Generate ~~Poisson~~-disk Sampling Patterns” SIGGRAPH 2006.



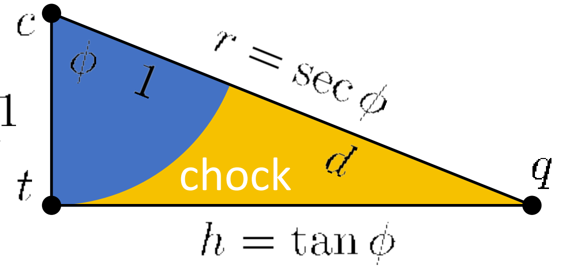
Advancing front

- Next disk distance [r, 2r] from a prior disk
- Inverse transform sampling
  - Given an area formula for a shape
  - Invert it to generate samples uniformly by area



DeterminsticMPS

- True Poisson-disk pattern
- Uniform-random by area of a chock[1]
  - Area formula
    - 1 trig function



$$A(\phi) = (\tan\phi - \phi)/2$$

  - Inverse area transform
    - 5 Newton’s iterations for 15 digits of accuracy
      - For higher precision, double precision each iteration
    - O(M(b) log b)

- Uniform-random by area from a sector
  - Area formula
    - Integral over 6 trig and 2 inverse trig functions
    - And 2 variations
  - Inverse area transform
    - Binary search over area integral
    - O(M(b) (b + log b)) × time for integral
      M(b) = time for b-bit multiplication

$A^{-1}(uA) =$ 5 Newton’s iterations

$$\phi_0 = \sqrt[3]{6uA}$$

$$\phi_{i+1} = \phi_i - \frac{\tan\phi_i - \phi_i - 2uA}{\tan^2\phi_i}$$

$$r_s = \sqrt{v\tan^2\phi_s + 1}$$

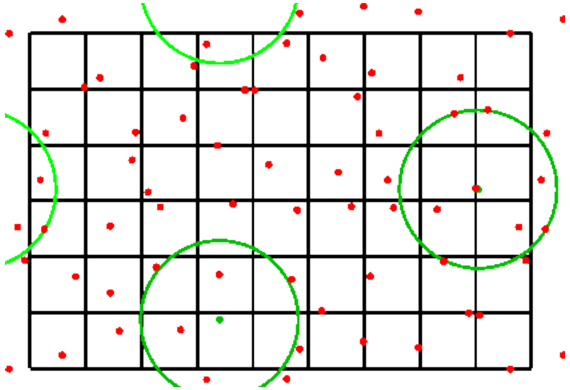
GridInner: Linear-time Poisson-disk patterns. Jones&Karger, 2011

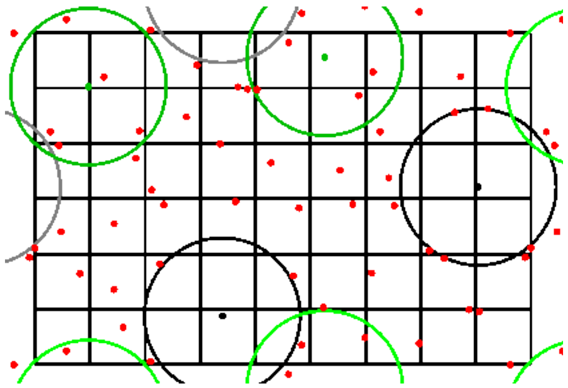
GridInner Algorithm

- Background grid
- Each square gets candidate sample
  - Random expovariate time
- Candidates earlier than neighbors **accepted**
  - In any order, congruent to Poisson-disk process
  - Covered neighbors are **resampled**
    - Uncovered scooped-square
    - Time += expovariate in remaining grid-square area

$$\mathrm{time}(x \in \Omega) = -\ln(v)/\mathrm{Area}(\Omega) \quad \text{random } v \in [0,1]$$

  - Repeat until domain is covered
- Congruent to Poisson-disk process





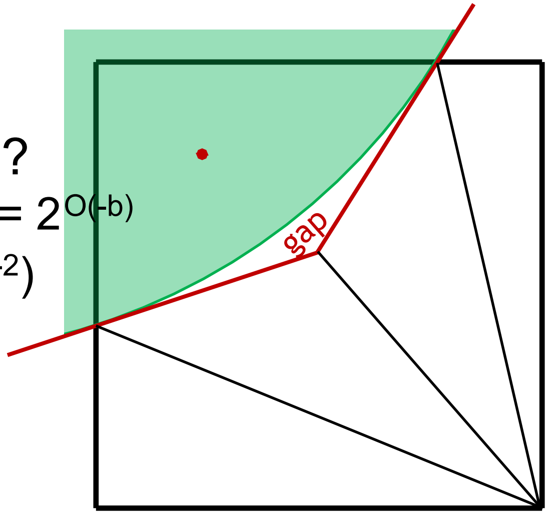
What's not to love?

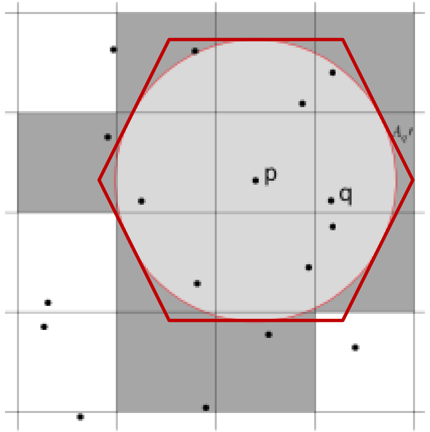
Resampling implementation

- Disks approximated by s-sided polygons
- Subtract polygon from square
- Triangulate
- Pick a triangle
- Pick sample in triangle

How many sides do we need?

- b-bits of accuracy, need gap = $2^{O(-b)}$
- gap = $1 - \cos(\pi/s) = O(\text{sides}^{-2})$
- sides = $2^{O(b)}$

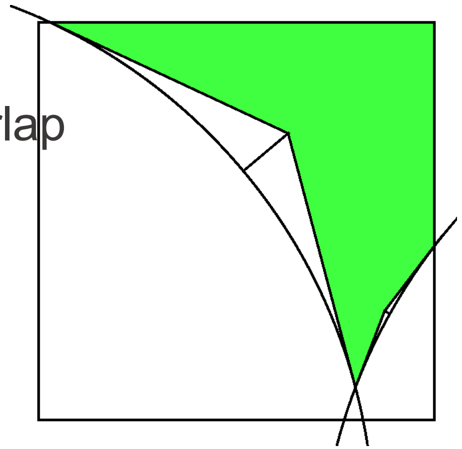


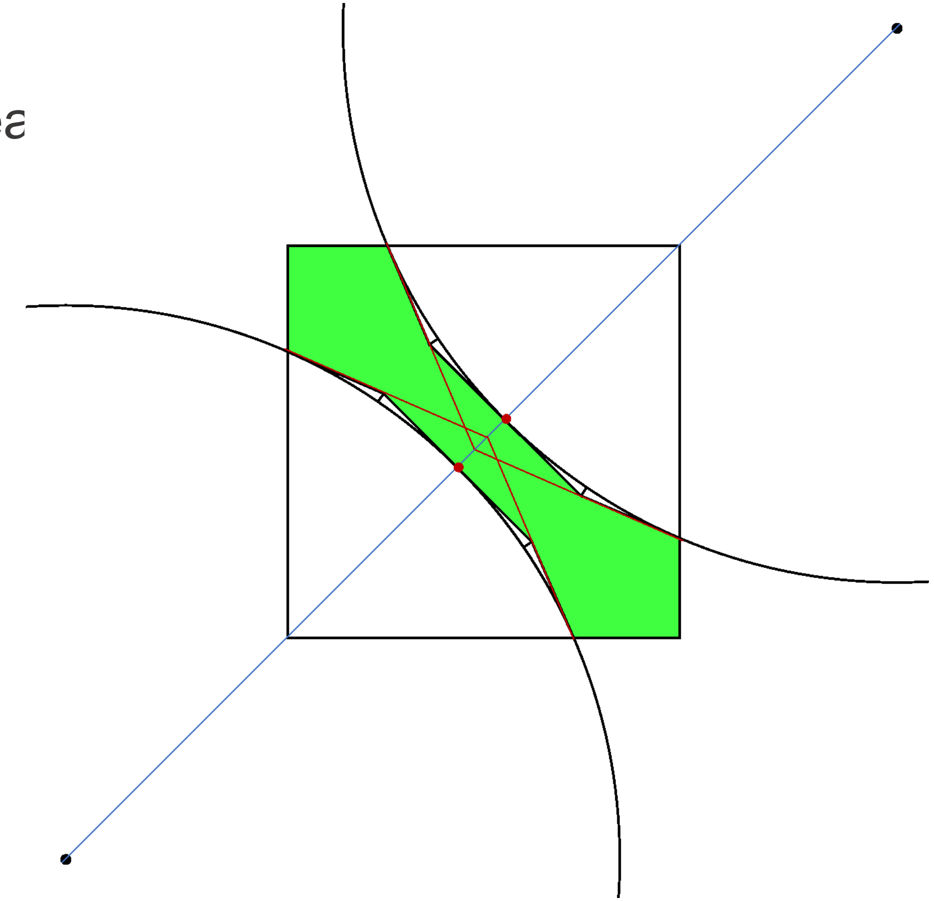


DeterministicMPS (this paper)

Fix: use chocks to exactly represent uncovered area

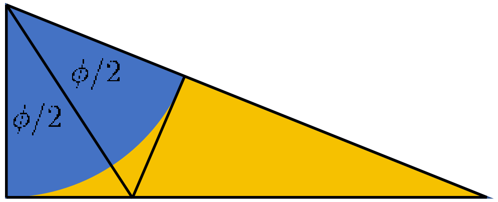
- Not polygonal approximation





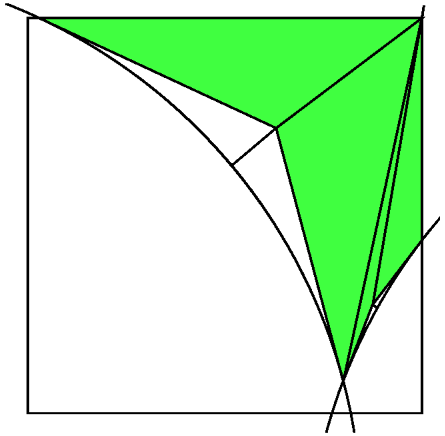
Trim chocks

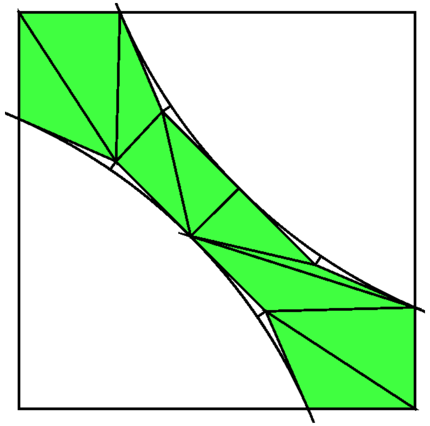
- Split to ensure they don’t overlap



Triangulate remainder

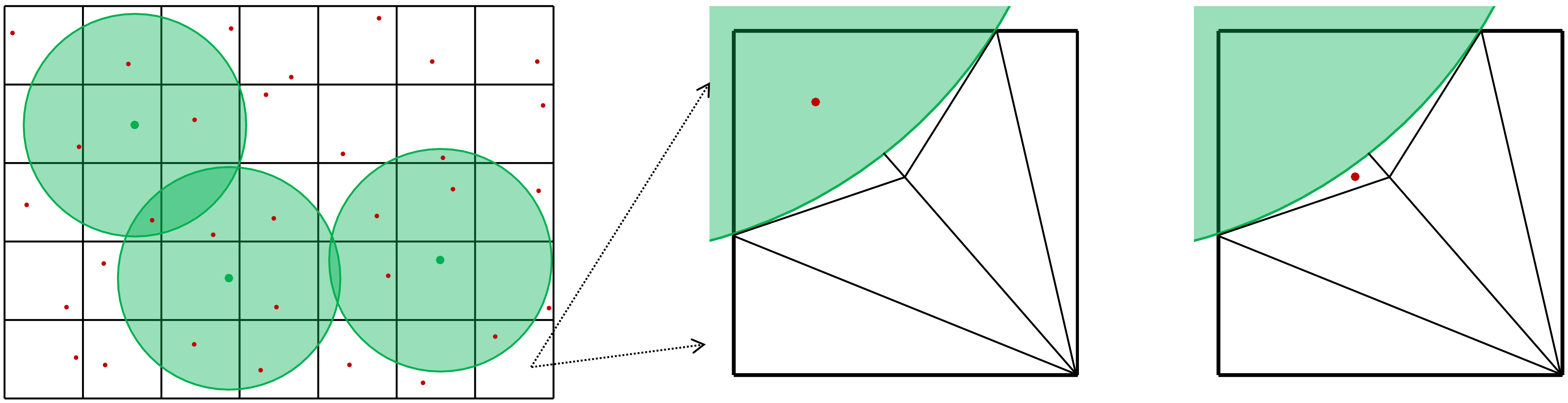
- Ear clipping





DetermisticMPS

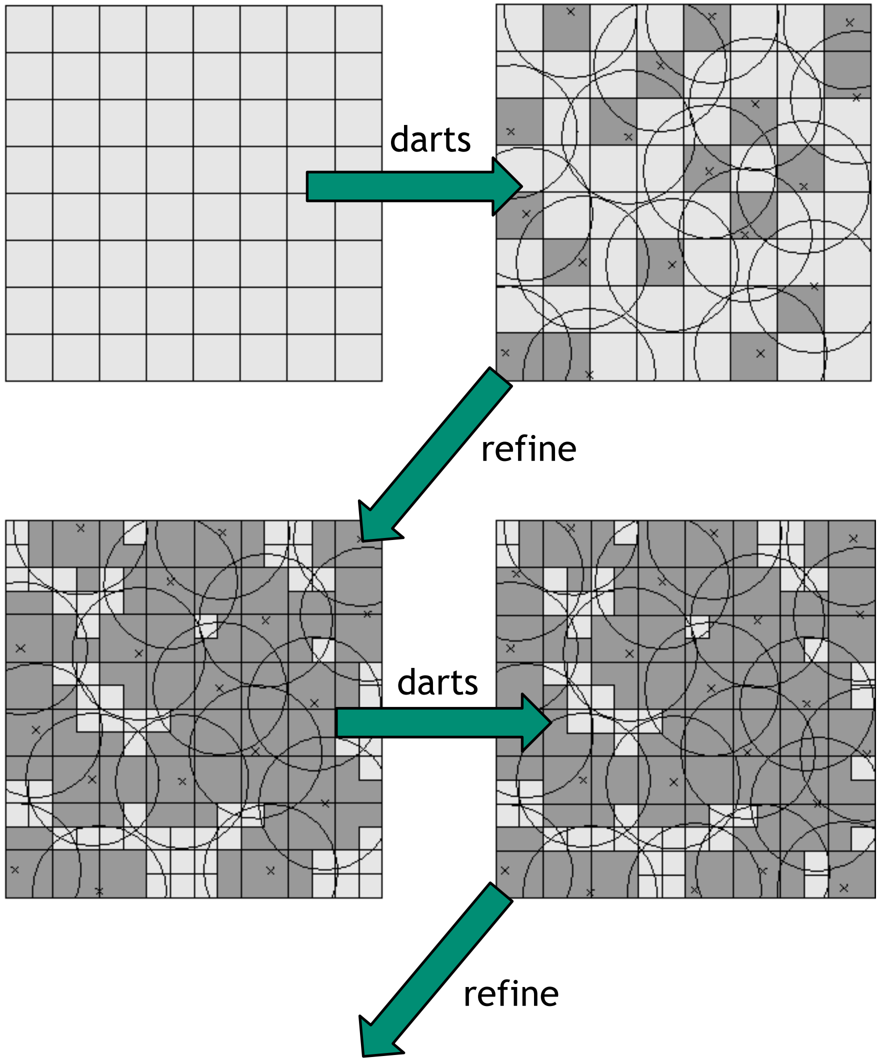
Resampling in constant time, exact up to numerical precision



Simple Maximal Poisson-Disk Sampling

Equivalent to dart-throwing process, identical output

- Guarantees saturation
- Runtime & memory, **empirically** $\mathbb{E}(n)$



Other extreme from DeterministicMPS

- Approximate uncovered regions by squares
- Geometric computations as coarse (and fast!) as possible
- Rejections occur

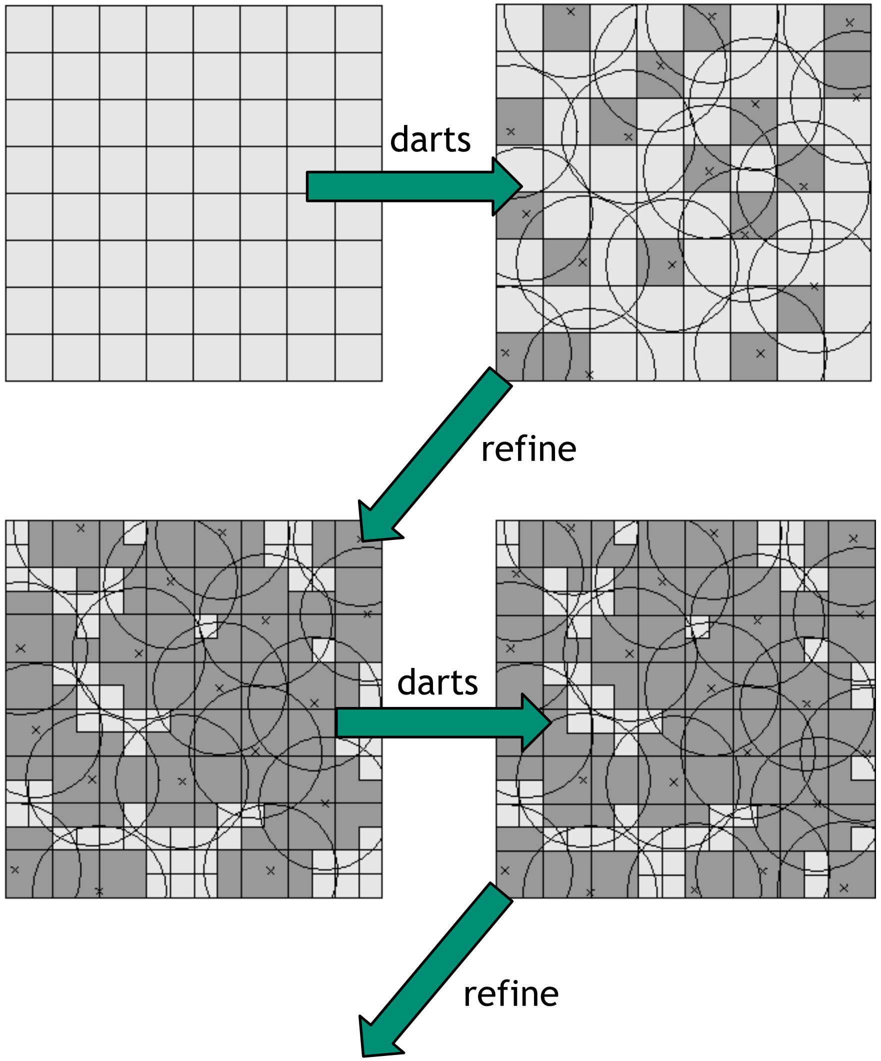
```
SimpleMPS Algorithm
divide domain into cubes
while ∃cubes, and diagonal > machine precision
  do (k × #cubes) times:
    pick a cube
    pick sample from cube
    if distance(sample, prior samples) > r
      accept sample
      discard cube
  refine cubes
  discard covered cubes
```

SimpleMPS, "A Simple Algorithm for Maximal Poisson-Disk Sampling in High Dimensions," Eurographi

Simple Maximal Poisson-Disk Sampling

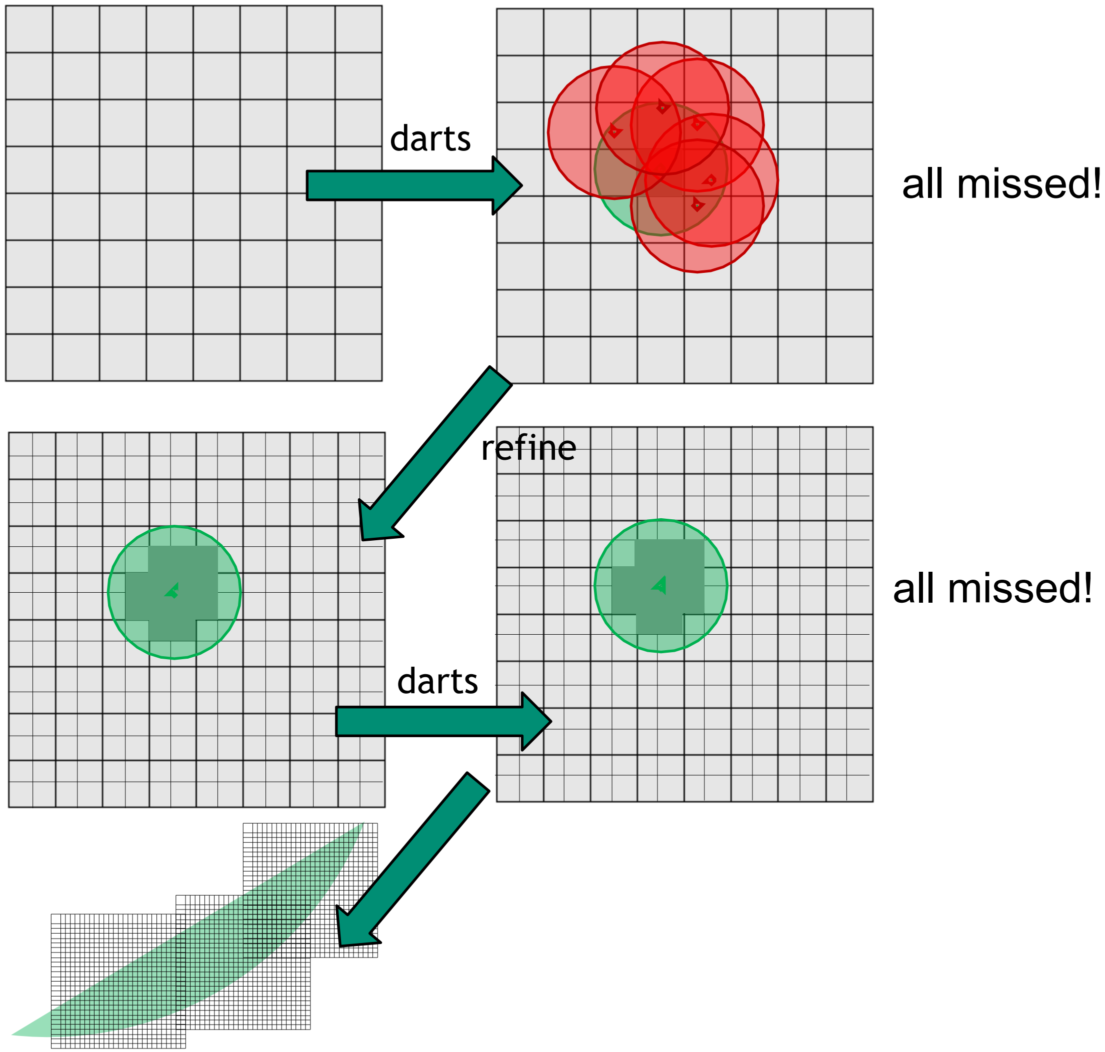
Equivalent to dart-throwing process, identical output

- Guarantees saturation
- Runtime & memory, **empirically** $\mathbb{E}(n)$



Worst case

$\mathrm{O}(n2^b)$



refine to machine precision, no more misses

DeterministicMPS relationship summary

method	approx.	Poisson-disks	maximal	expected time	deterministic time	precision complexity
Dart-throwing [DW85]	outer	Y	-	*1*	-	1
Scallop-MDS [DH06a]	in	-	Y	-	$n \log n$	$b + \log b$
Voronoi-MPS [Jon06]	outer	Y	Y	$n \log n$	-	1
GridOuter-MPS [EPM□11]	outer	Y	Y	$n \log n$	-	1
Sequential-MPS [ours]	-	Y	Y	-	$n \log n$	$\log b$
Simple-MPS [EMP□12]	outer	Y	Y	$n^{\dagger}$	-	2^b
GridInner-MPS [JK11]	in	$Y^{\ddagger}$	$Y^{\ddagger}$	-	n	2^b
ChockSubdivision-MPS [ours]	-	Y	Y	-	n	$b + \log b$
Deterministic-MPS [ours]	-	Y	Y	-	n	$\log b$

GridInner

- Samples from subset of uncovered
  - Disk-square geometry construction
  - Triangulate
- No rejections
- Exponential complexity in accuracy of subset matching true domain and true Poisson-disk sampling

ScallopMDS

- Samples from subset of uncovered
  - Subset never matches true domain
  - Not Poisson-disk sampling
- No rejections
- Inverse area transform
  - Linear complexity in numerical accuracy

Read paper, Background other MPS

It's excellent, but don't take my word fo

take the reviewers ☺

DeterministicMPS

- No rejections
- Samples exactly from uncovered
- Log complexity in numerical accuracy

SimpleMPS

- Samples from superset of uncovered
- Rejections occur
- Exponential worst-case

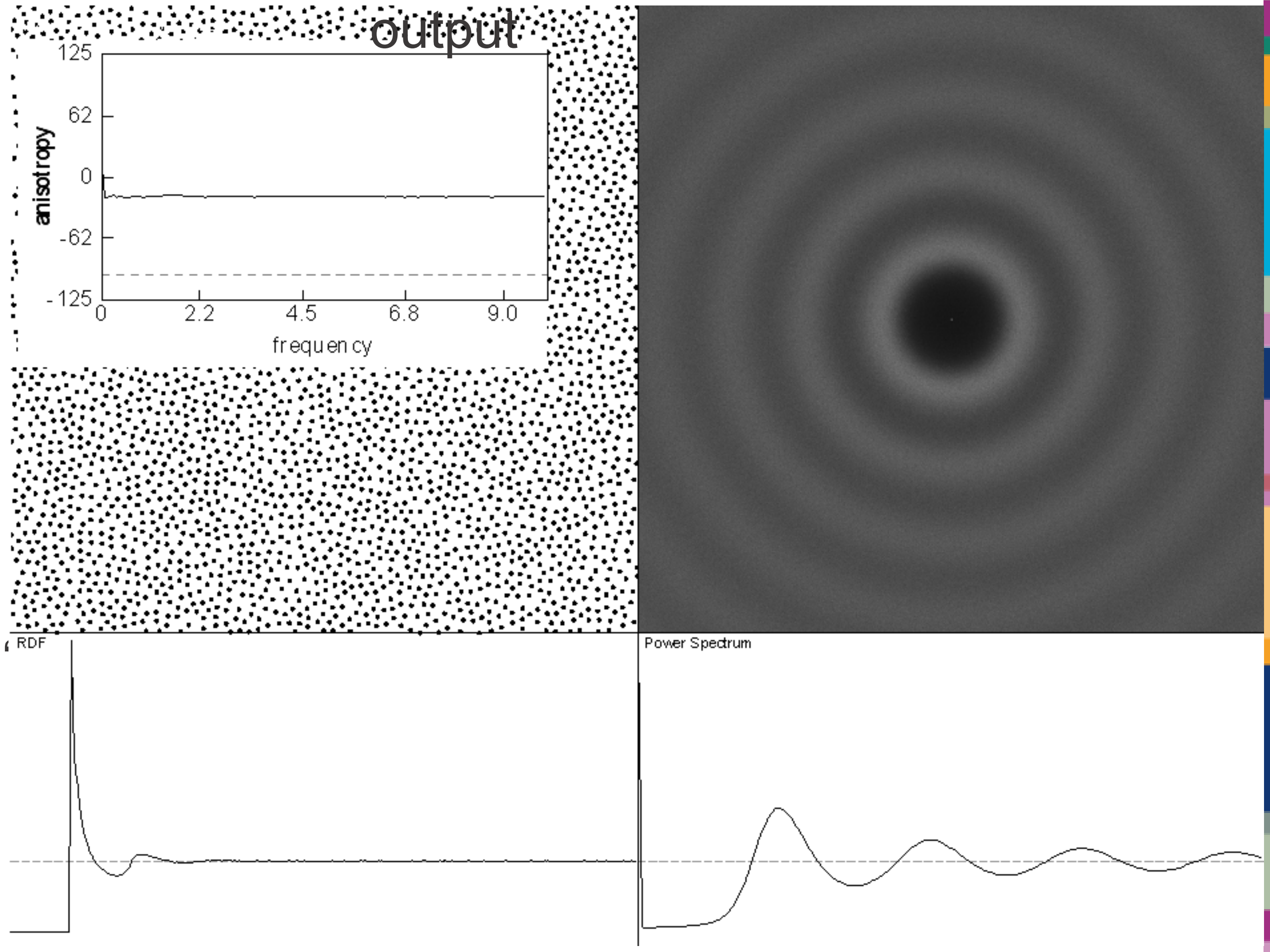
Practical Matters

Efficiency in practice
- 100k samples / second
  - Order of magnitude faster than all other maximal Poisson-disk methods compared
  - except SimpleMPS is 2-3× faster!

Open source
- https://github.com/samitch/DeterministicMPS
- Free for any use
- No dependencies. No libraries. Nothing. Just say
- Replicates paper figures

Code size
- Relatively large due to geometric computations
- Chock sampling itself small, few dozen lines.

Verified true MPS output



Open Problems

DeterminsticMPS

- Embedded surfaces
- 3D
- Need new inverse area transforms

Can you prove

- Throw darts into a domain, then constant fraction accepted?
- Serial: SimpleMPS provable expected runtime
- Parallel: PixelPie provable expected constant

Step blue-noise and non-uniform distributions

- Currently
  - Posteriori optimization, slow
  - Priori placement
    - Use chocks to avoid rejections and approximations
    - Complexity?
      - Quickly varying distribution = increased #neighbors

Thank you!

“Il semble que la perfection soit atteinte non quand il n'y a plus rien à ajouter, mais quand il n'y a plus rien à retrancher.”
“Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.”

– **Antoine de Saint-Exupéry,** **Airman's Odyssey**

Open source

- https://github.com/samitch/DeterministicMPS
- Free for any use