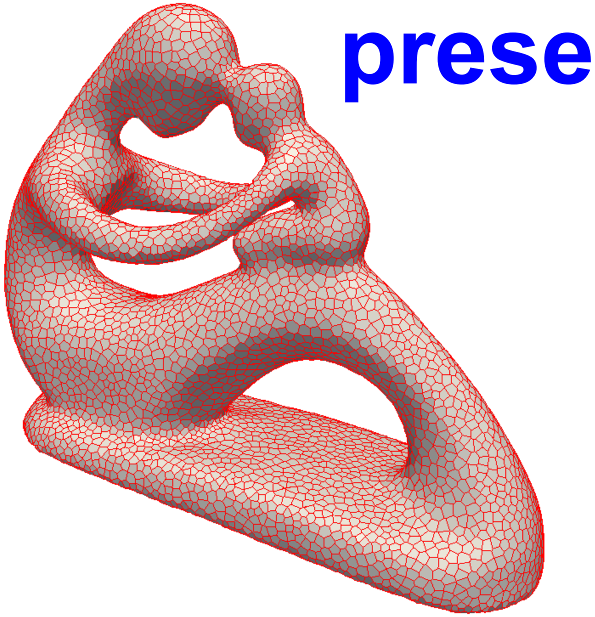
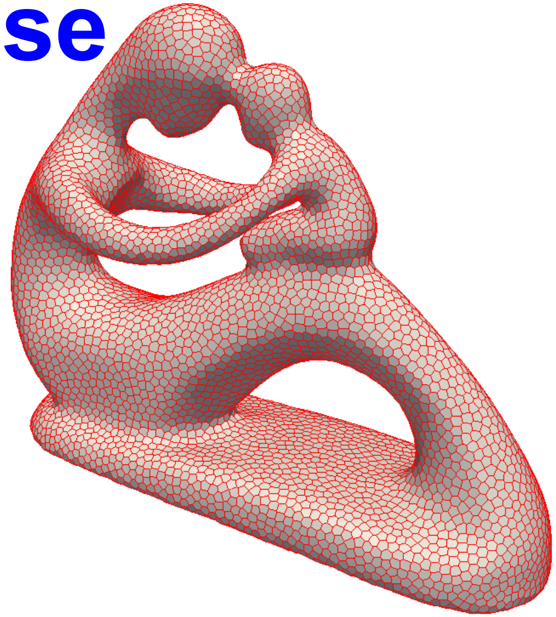
Improving spatial coverage while preserving blue noise





Mohamed S. Ebeida[1]

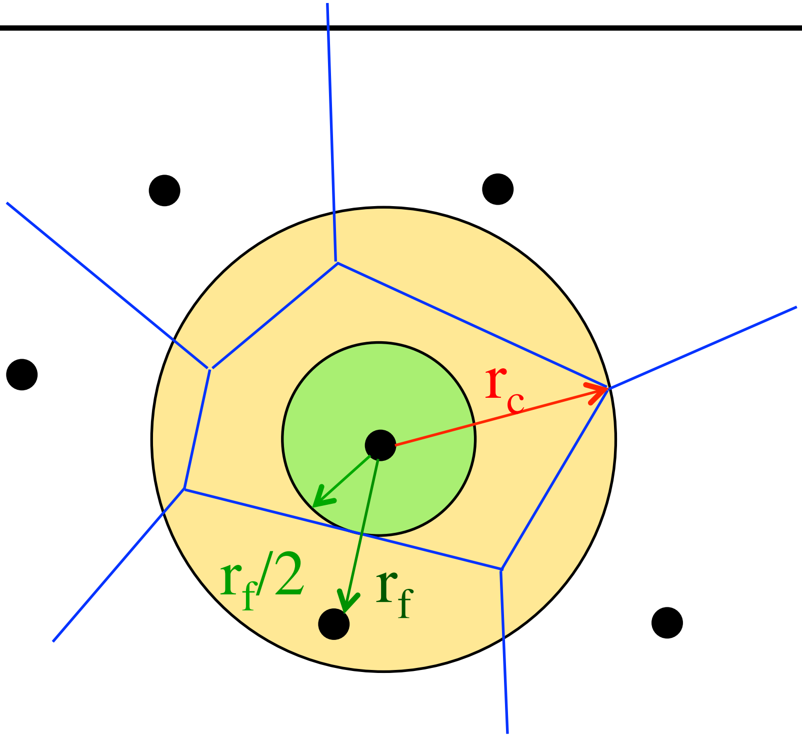
M. A. Awad[2], X. Ge[3], A. H. Mahmoud[2], S. A. Mitchell[1], P.M. Knupp[1], and L. -Y. Wei[4]

[1]Sandia National Laboratories, [2]Alexandria University,
[3]Ohio-State Univ, [4]University of Hong Kong

Siam Conference on Geometric and Phsyical Modeling

November, 13th 2013

Point Sets: Well-spaced



r_{c} = coverage
farthest distance from domain point to sample point

r_{f} = free
shortest distance between sample points

- Well-spaced =
  - Farthest Voronoi vertex (coverage) not much farther than closest Delaunay neighbor (free)
  - Measured by Voronoi cell aspect ratio (A) or beta = r_c / r_f
  - beta <= 1 is often the goal for uniform distributions

Point Sets: Random

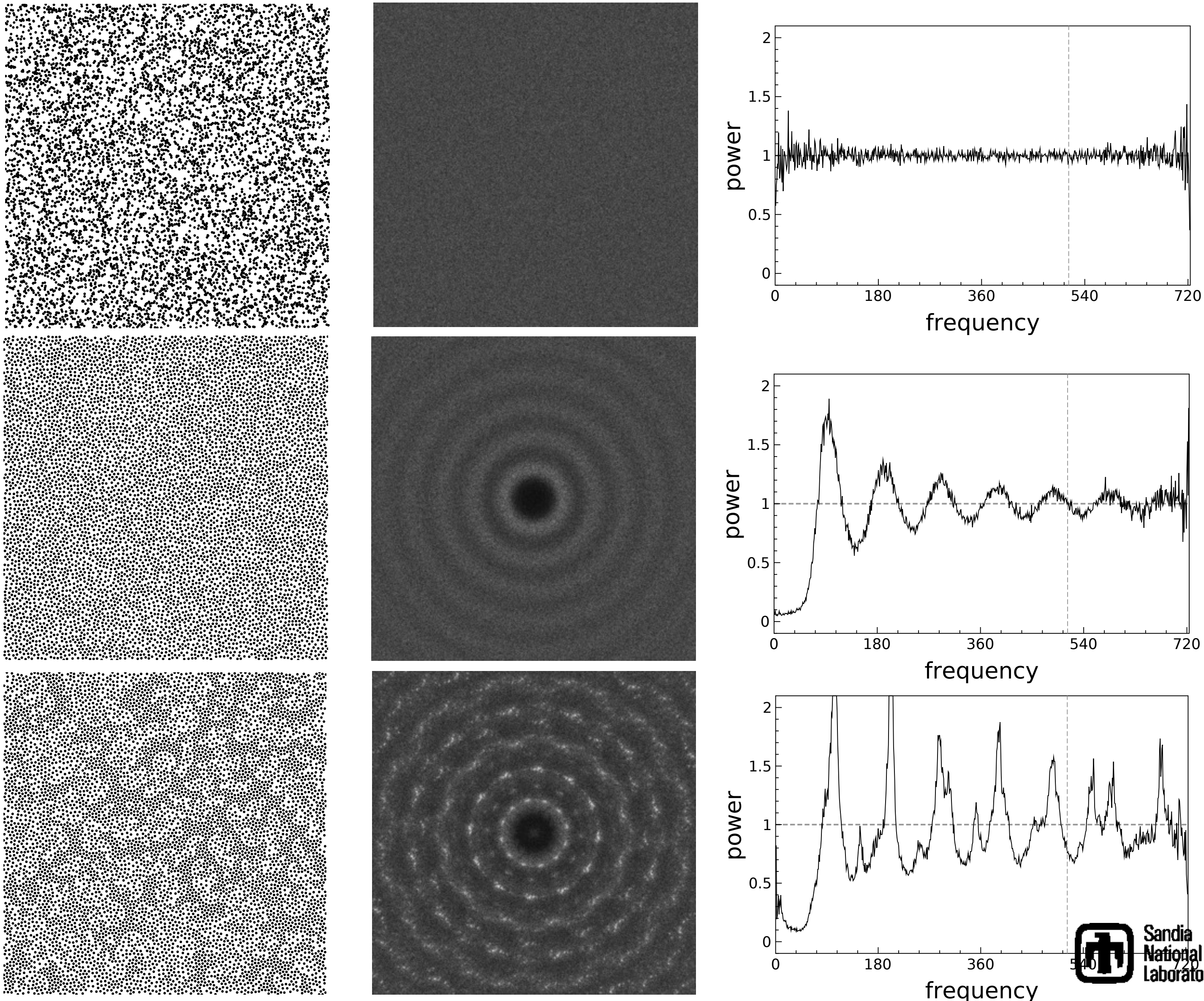
- Random with no constraints (white noise)

- Random with minimum separation (blue noise)

  $r_f = r_c$

- Correlated Points

  $r_f = r_c$

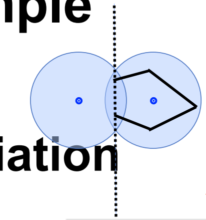


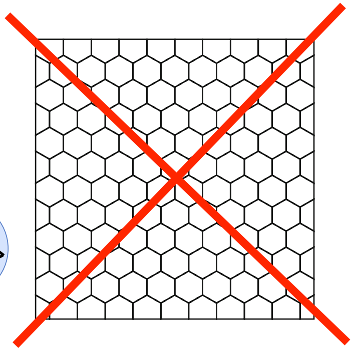
Why Do We Care?!

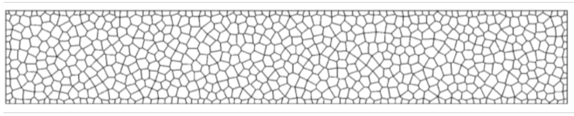
Applications for Random Well Spaced point Sets

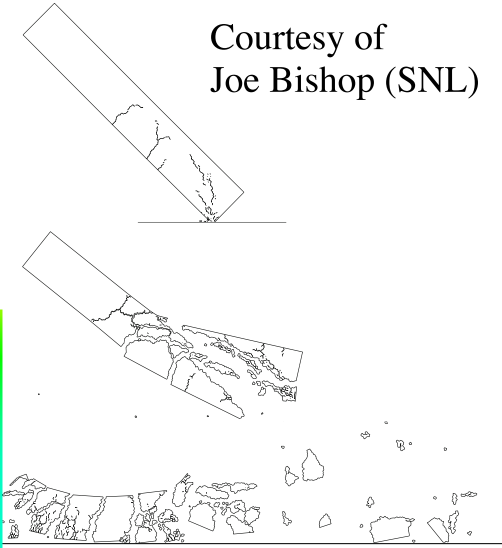
Provably Good Meshing

- **Physics simulations**
  - **Voronoi mesh, cell = points closest to a sample**
  - **Fractures occur on Voronoi cell boundaries**
    - **Mesh variation models material strength variation**
    - **CVT, regular lattices give unrealistic cracks**
      - **Unbiased sampling gives realistic cracks**
  - **Ensembles of simulations**
  - **Domains: non-convex, internal boundaries**

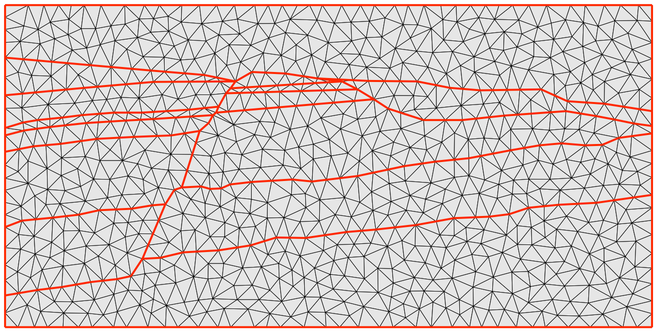




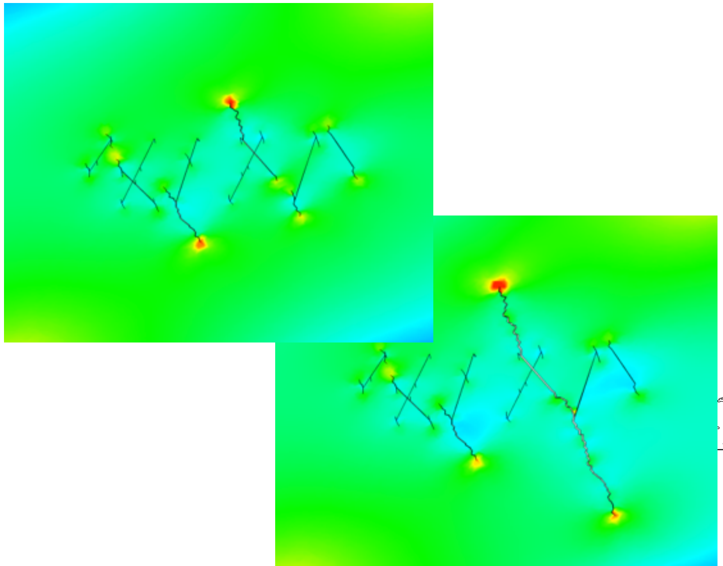




Courtesy of
Joe Bishop (SNL)



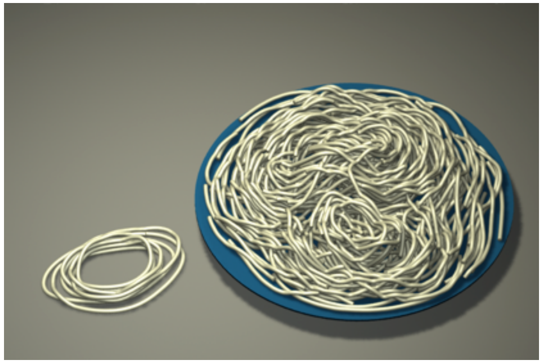
Seismic Simulations
maximal helps Δ quality



Sandia National Laboratories

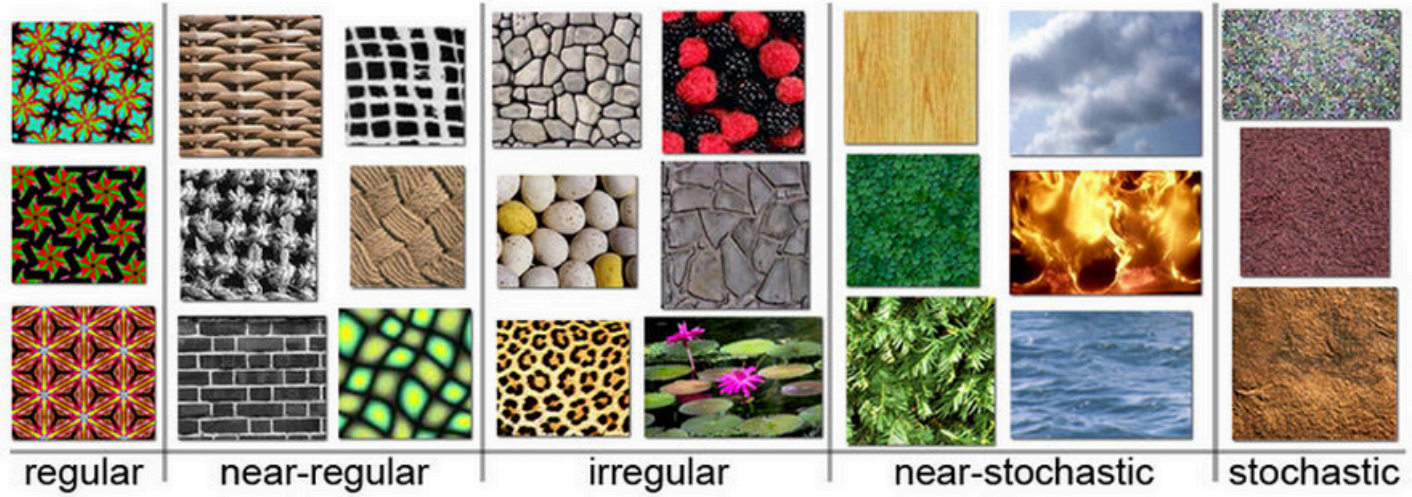
Motivating from Modern Graphics: Texture Synthesis

- Real-time environment exploration. Games! Movies!
- Algorithm to create output image from input sample
  - Arbitrary size
  - Similar to input
  - No visible seams, blocks
  - No visible, regular repeated patterns

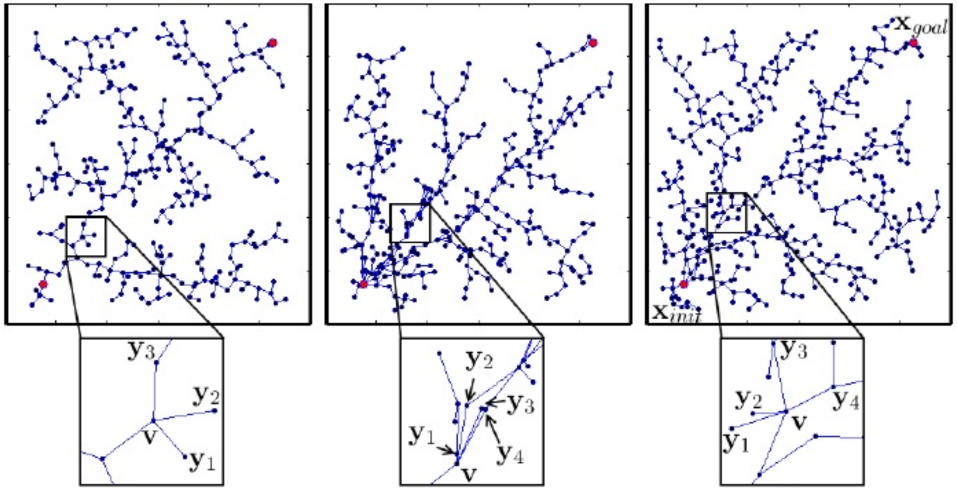


Spaghetti
Li Yi Wei
SIGGRAPH 2011

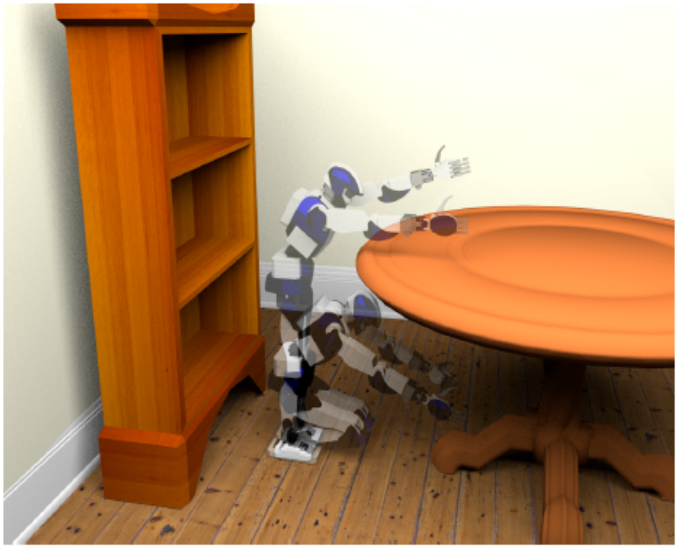
examples from wikipedia:



Robot Motion Planning



(a) RRT (b) Parallel RRT (c) Our Algorithm



Real time motion planning 23 DOF

Precomputed Well-Spaced points directs parallel tree expansion and enables real-time motion planning in higher dimensions

Pictures and results provided by: Chonhyon Park (Dinesh's group)

That was the applications!

…

Now for the algorithms!

How can we generate a random well spaced point set?

… So We Need to Generate Random Well-Spaced Points. But How?!! → Maximal Poisson-Disk Sampling (MPS)

- What is MPS?
  - Insert random points into a domain, build set X

Disk-free condition

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

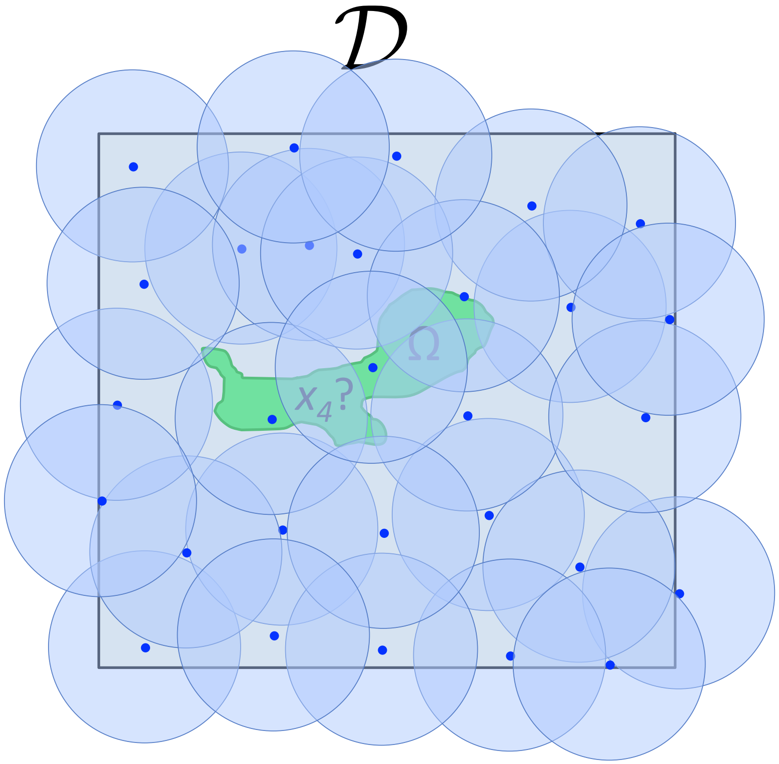
Bias-free condition

$$\forall x_i \in X, \forall \Omega \subset \mathcal{D}_{i-1} :$$

$$P\left(x_i \in \Omega\right) = \frac{\text{Area}(\Omega)}{\text{Area}(\mathcal{D}_{i-1})}$$

Maximal condition

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



Simple Problem?!

My initial thoughts (2010):

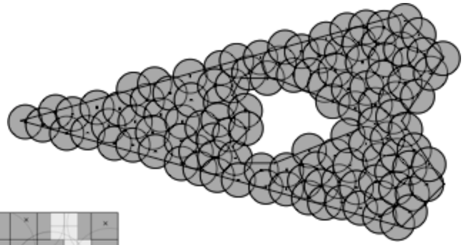
Generate a bunch of disks in a box, sounds too simple with minor impact!

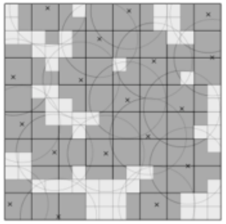
Probably I will forget about it after this project.

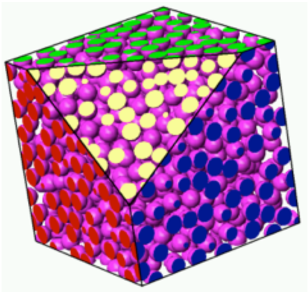
I was Completely wrong!

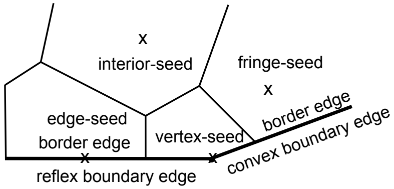
Main Published Results

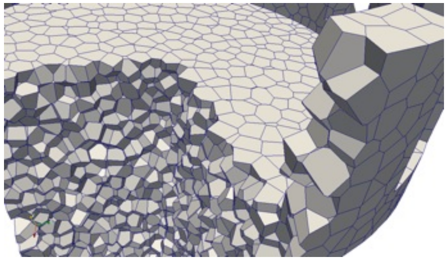
- First E(n log n) algorithm with provably correct output
  - Efficient Maximal Poisson-Disk Sampling,
    Ebeida, Patney, Mitchell, Davidson, Knupp, Owens,
    SIGGRAPH 2011
- Simpler, less memory, provably correct, faster in practice but no run-time proof
  - A Simple Algorithm for Maximal Poison-Disk Sampling in High Dimensions,
    Ebeida, Mitchell, Patney, Davidson, Owens
    Eurographics 2012
- Voronoi Meshes
  - Sites interior, close to domain boundary are OK, not the dual of a body-fitted Delaunay Mesh
  - Uniform Random Voronoi Meshes
    Ebeida, Mitchell
    IMR 2011
- Delaunay Meshes
  - Protect boundary with random balls
  - Efficient and Good Delaunay Meshes from Random Points
    Ebeida, Mitchell, Davidson, Patney, Knupp, Owens
    SIAM GD/SPM 2011 → Computer Aided Design
- MPS with varying radii
  - Adaptive and Hierarchical Point Clouds
  - Variable Radii Poisson-disk sampling
    Mitchell, Rand, Ebeida, Bajaj
    CCCG 2012

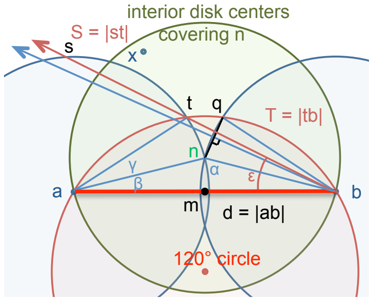






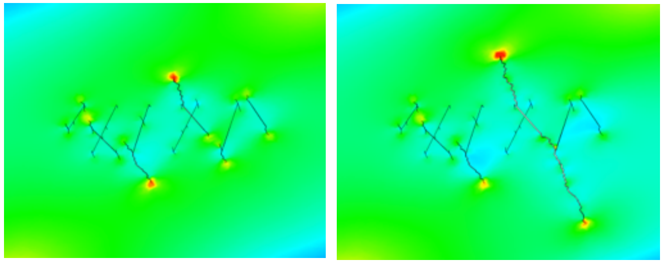


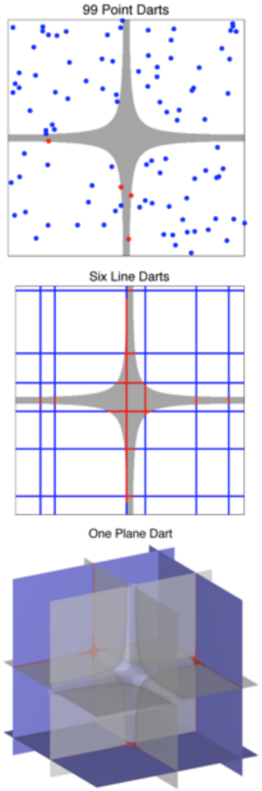


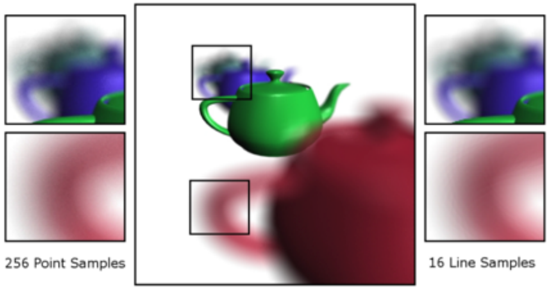


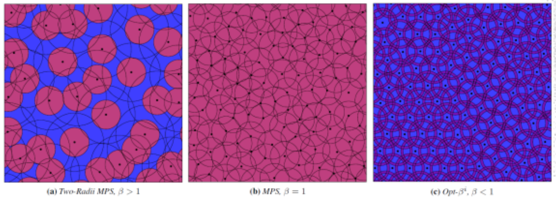
Main Published Results

- Simulation of Propagating fractures
  - Mesh Generation for modeling and simulation of carbon sequestration processes
    Ebeida, Knupp, Leung, Bishop, Martinez
    SciDAC 2011
- Hyperplanes for integration, MPS and UQ
  - K-d darts,
    Ebeida, Patney, Mitchell, Dalbey, Davidson, Owens,
    TOG “to appear”
- Rendering using line darts
  - High quality parallel depth of field using line samples,
    Tzeng, Patney, Davidson, Ebeida, Mitchell, Owens
    HPG 2012
- Reducing Sample size while respecting sizing function
  - A simple algorithm that replaces 2 disks with one while maintaining coverage and conflict conditions
  - Sifted Disks
    Ebeida, Mahmoud, Awad, Mohammad, Mitchell, Rand, Owens
    EG 2013
- MPS with improved Coverage
  - Using $r_c < r_f$
  - Improving spatial coverage while preserving blue noise
    Ebeida, Awad, Ge, Mahmoud, Mitchell, Knupp, Wei
    SIAM GD/SPM 2013 → Computer Aided Design





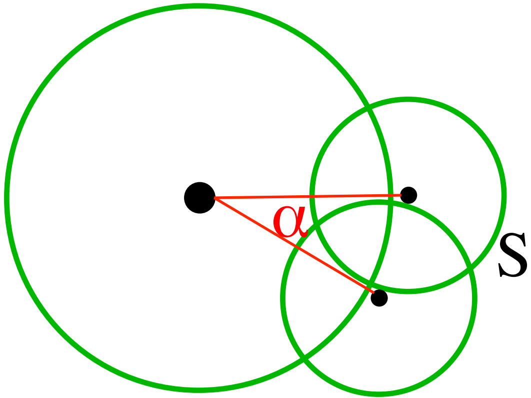




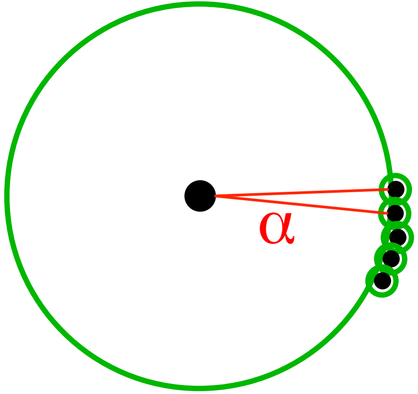
(a) Two-Radii MPS, $\beta > 1$ (b) MPS, $\beta = 1$ (c) Opt-β^*, $\beta < 1$

Prior Results: CCCG'11
How fast can radii vary?

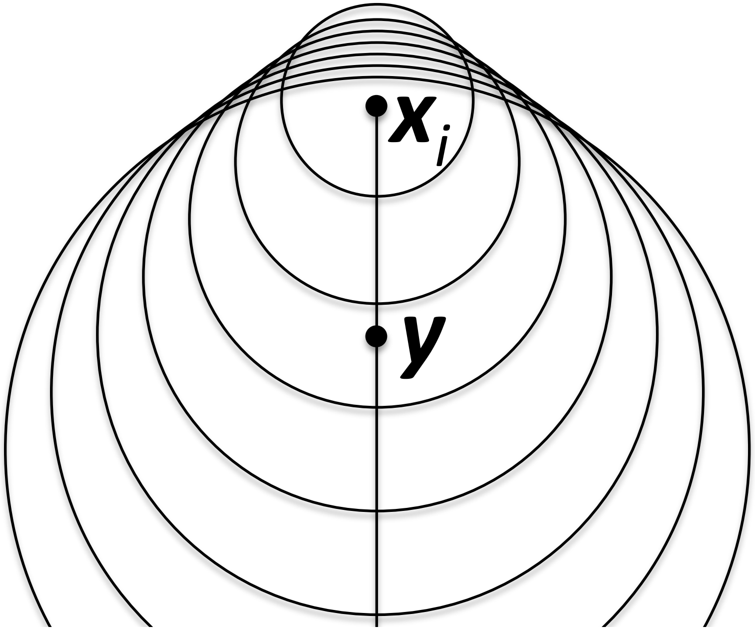
- If varies slowly
  - bounded # neighbors for disk conflict checks <-> bounded-angle DT
- If shrink too fast
  - Unbounded # neighbors
  - Infinite run-time
  - Zero angles in triangulation



Still big, OK



Vanishing radius, infinite neighbors

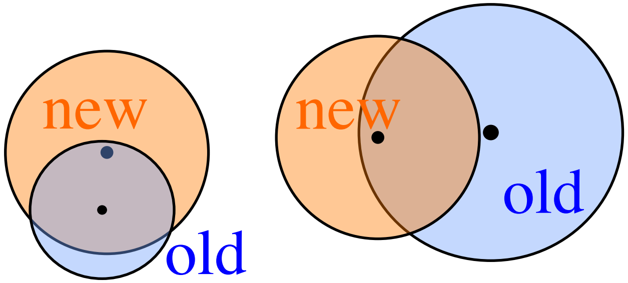


Q. How fast can it vary?
A. Depends how Conflict is defined.

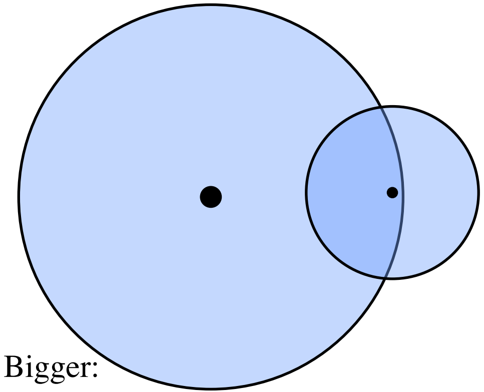
L is Lipschitz constant: f(x)-f(y) < L |x-y|

Four common methods in Graphics

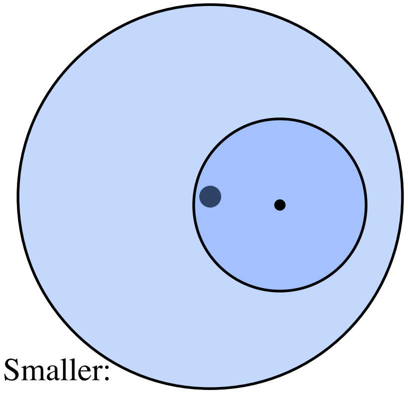
Method	Distance Function	Order Independent	Full Coverage	Conflict Free	Edge Min	Edge Max	Sin Angle Min	Max L
Prior	$r(\mathbf{x})$	no	no	no	$1/(1+L)$	$2/(1-2L)$	$(1-2L)/2$	$1/2$
Current	$r(\mathbf{y})$	no	no	no	$1/(1+L)$	$2/(1-L)$	$(1-L)/2$	1
Bigger	$\max(r(\mathbf{x}), r(\mathbf{y}))$	yes	no	yes	1	$2/(1-2L)$	$(1-2L)/2$	$1/2$
Smaller	$\min(r(\mathbf{x}), r(\mathbf{y}))$	yes	yes	no	$1/(1+L)$	$2/(1-L)$	$(1-L)/2$	1



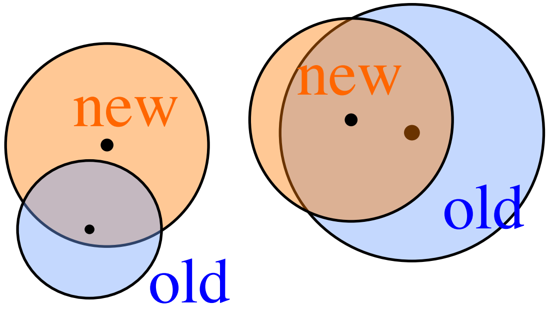
Prior:
new candidate disk center inside an old prior disk



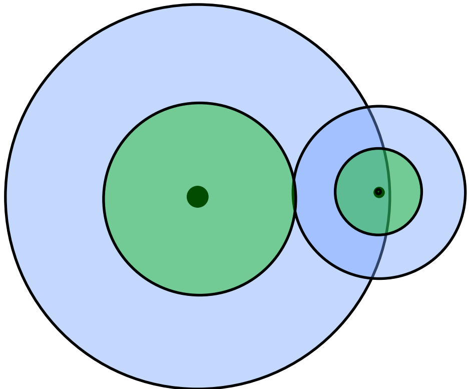
Bigger:
small disk center inside big disk center



Smaller:
big disk center inside small disk center



Current:
old prior disk center inside a new candidate disk



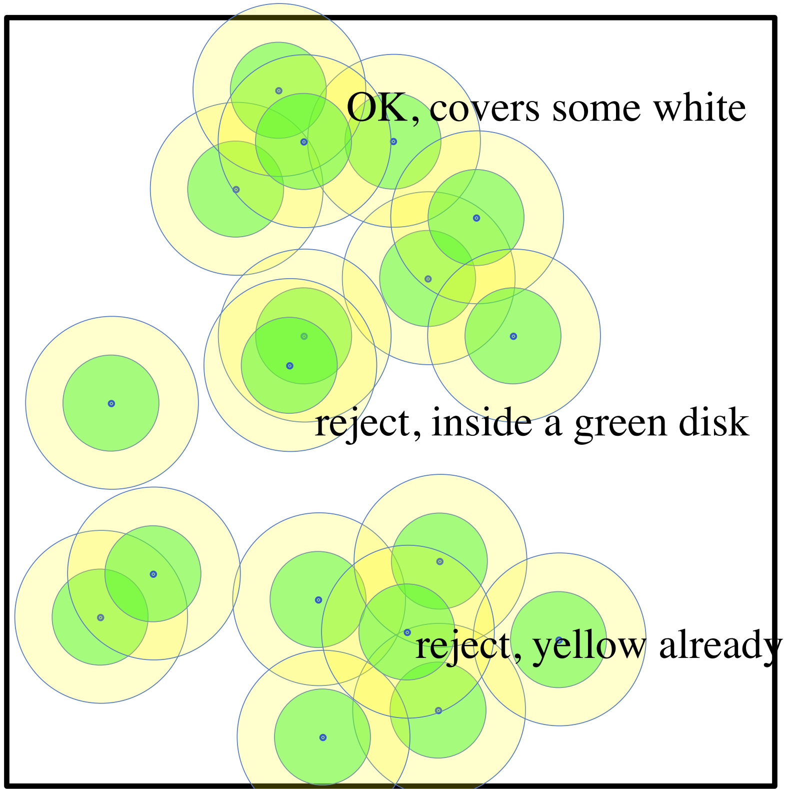
Bigger is stricter than
Sphere packing:
½ radius disks overlap
distance: sum(r(x),r(y))/2

Prior Results: CCCG'11
Decoupling coverage and conflict-free radii

- **Disk coverage radius larger than free radius**

  $R_c > R_f$ (yellow > green)

- **New disks must cover some unique uncovered area**
  - **Else maximal (limit) distribution would be the same**
  - **Contrast to Hard-core Strauss disc process: coverage disks are observed, no effect on process**



OK, covers some white

reject, inside a green disk

reject, yellow already covered

Process:

New candidate point uniform at random

(f) Rejected if center inside a small green disk

(c) Accepted if its yellow disk covers some white a

Alg:

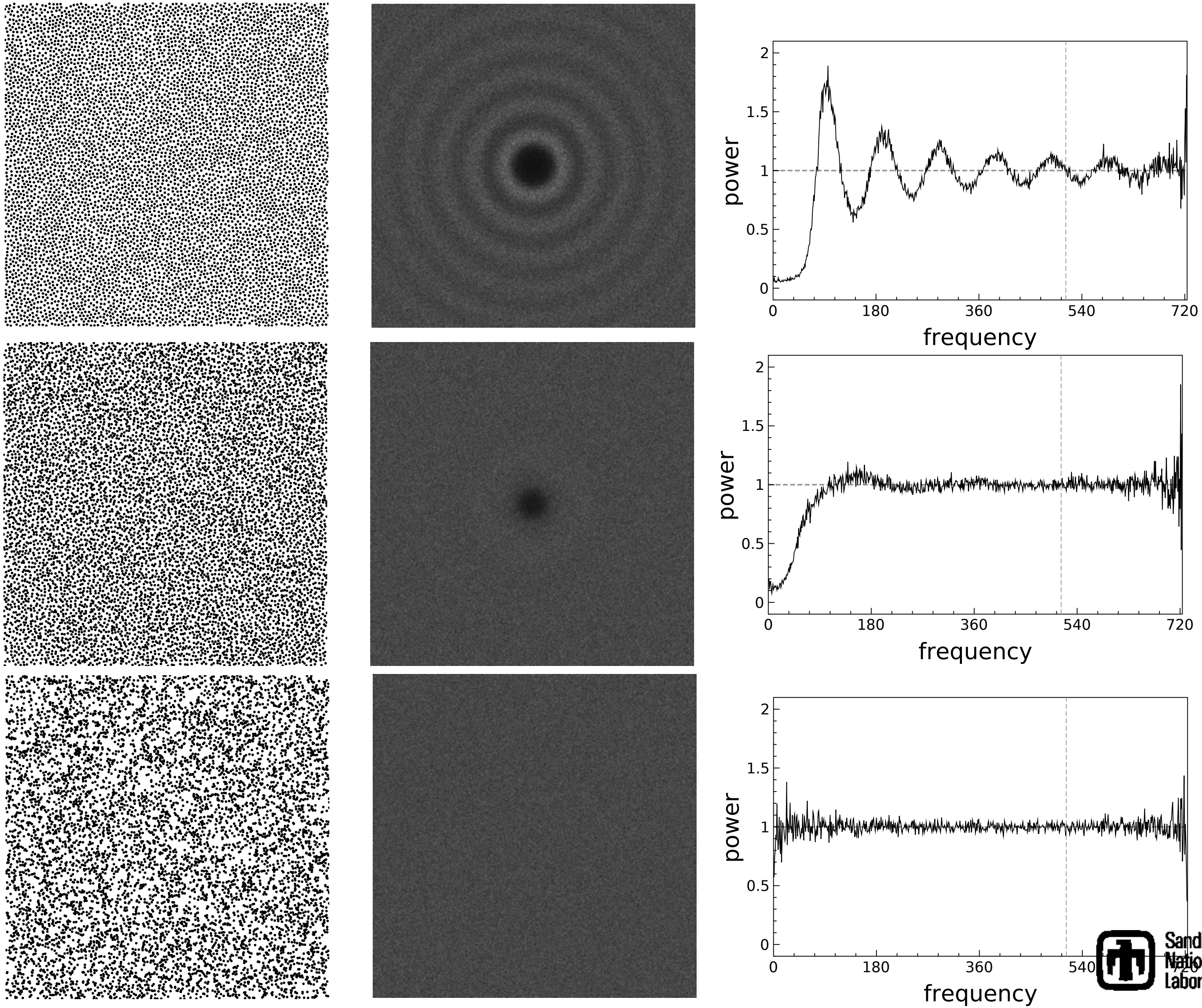
Only generate points in an outer approximation to regions satisfying (c) and (f) in the first place.

Two-radii MPS output

- **Classic MPS**
  $R_f = R_c$

- **Two-radii MPS**
  $2 R_f = R_c$
  - **R_f = min center dist**
  - **R_c =max Vor dist**

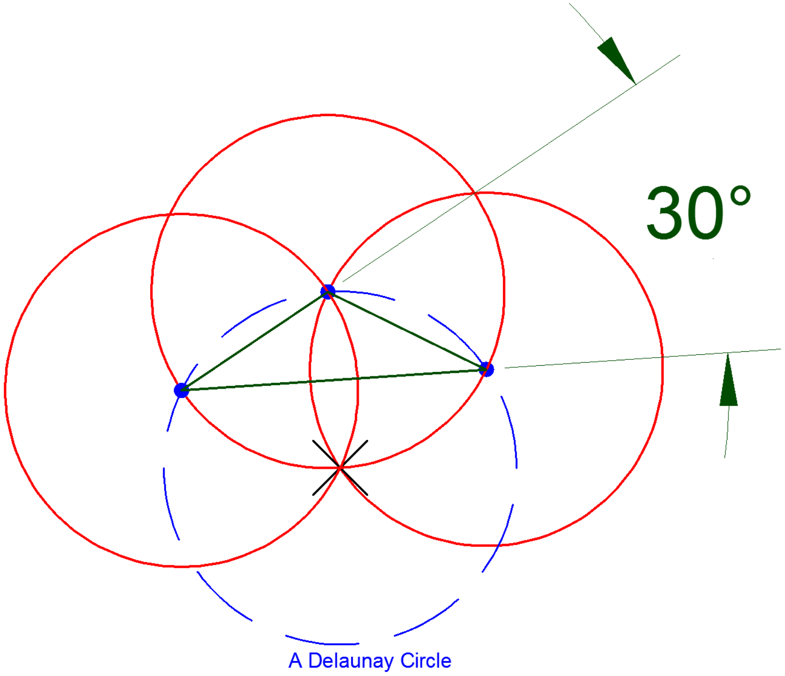
- **Uniform**
  R = 0
  non-maximal

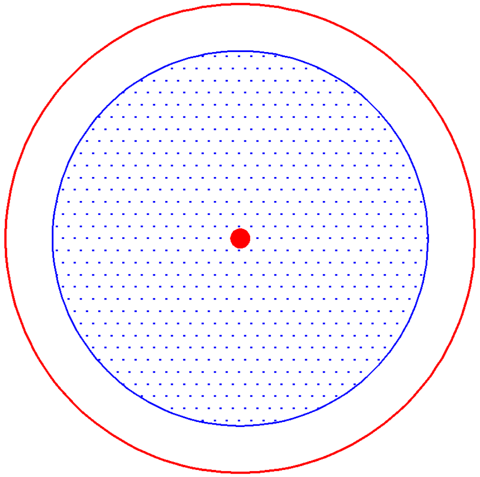


This Research: Improving spatial coverage

- $r_c > r_f$ increase randomness while degrading mesh quality

- Here we try the opposite direction $r_f < r_c$

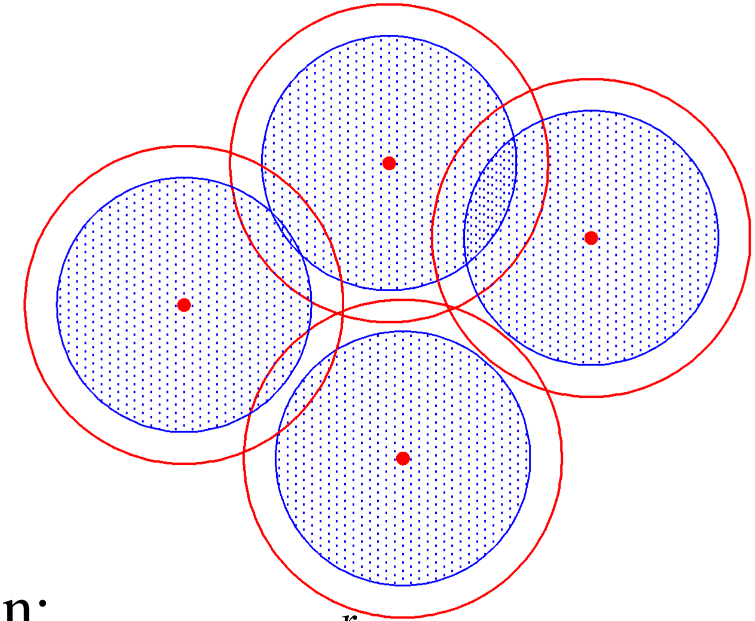




Question:
Can we decouple coverage radius from disk-free radius such r_c < r_f ?

Current Research: Sparse MPS (under preparation)

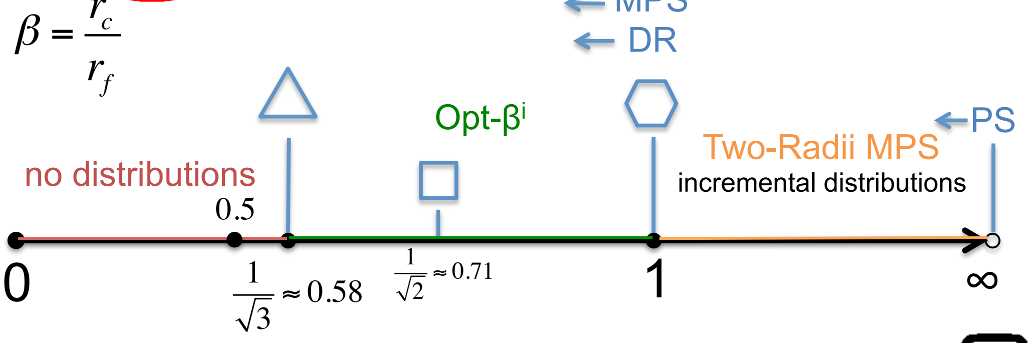
Answer: Yes, If the algorithm doesn't lead to configurations
Like this.



Unfortunately!
Vey hard to achieve
via sampling due to
global constraints

Impact of solution:
A tune-up parameter
to trade randomness
For better space coverage
e.g. better mesh quality

$$\beta = \frac{r_c}{r_f}$$



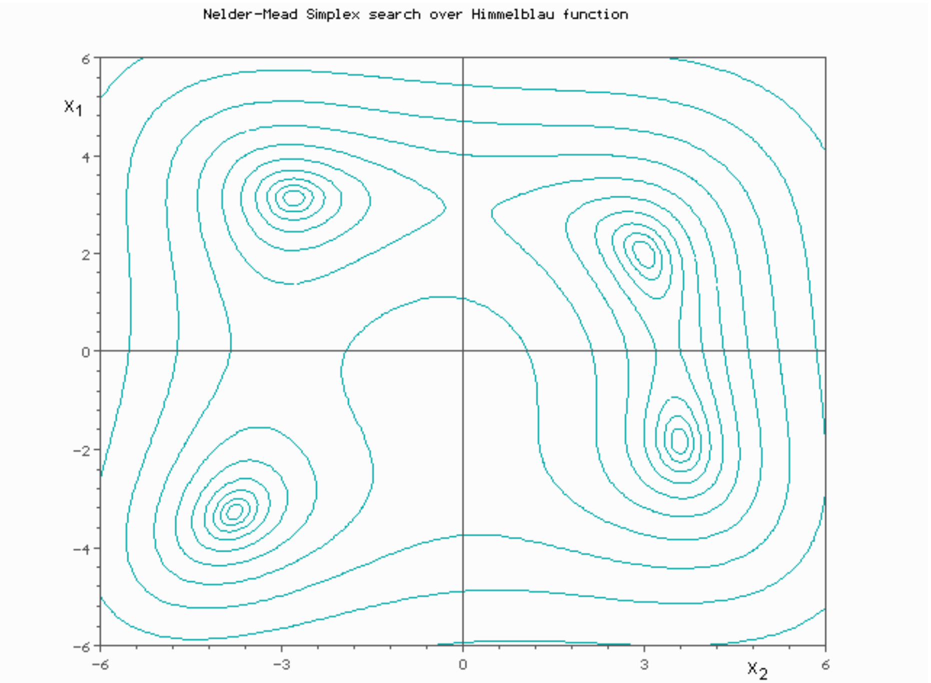
← MPS
← DR
←PS
Opt-β^i
Two-Radii MPS
incremental distributions
no distributions
0.5
0
$\frac{1}{\sqrt{3}} \approx 0.58$
$\frac{1}{\sqrt{2}} \approx 0.71$
1
∞

Our poor-man's solution

- Generate an MPS as usual and relocate points to optimize beta directly using Nelder Mead

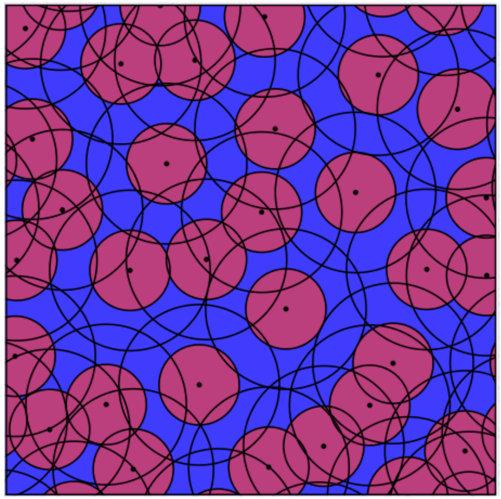
Four rules for relocating a point:

1. Reflection
2. Expansion
3. Contraction
4. Reduction

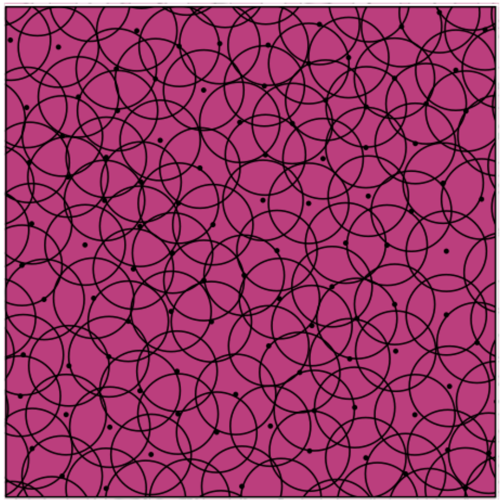


(c) P.A. Simionescu 2006

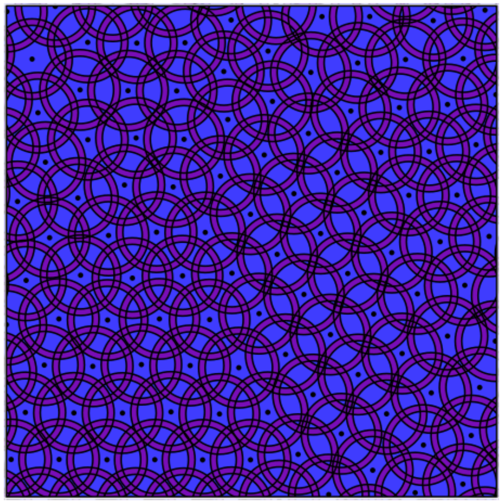
Results



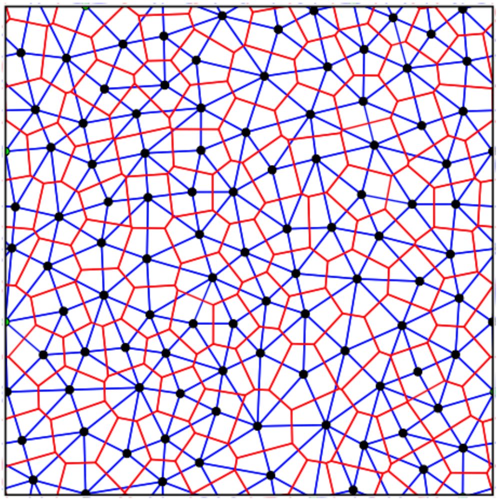
(a) Two-Radii MPS, $\beta > 1$



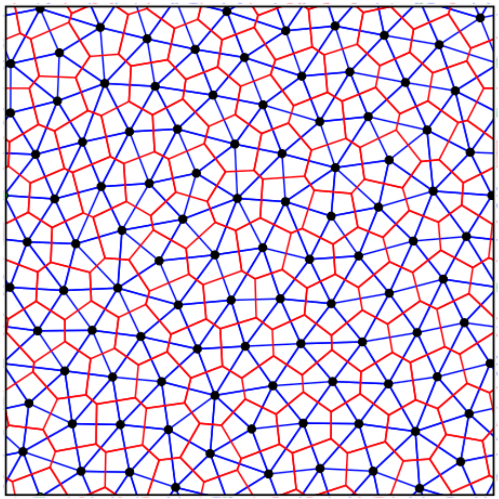
(b) MPS, $\beta = 1$



(c) Opt-β^i, $\beta < 1$

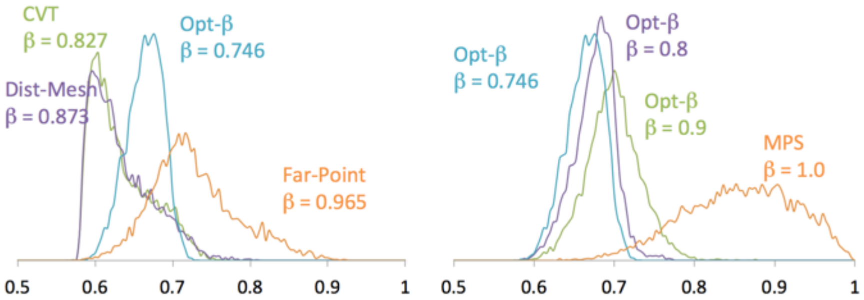


(d) MPS mesh, $\beta = 1$

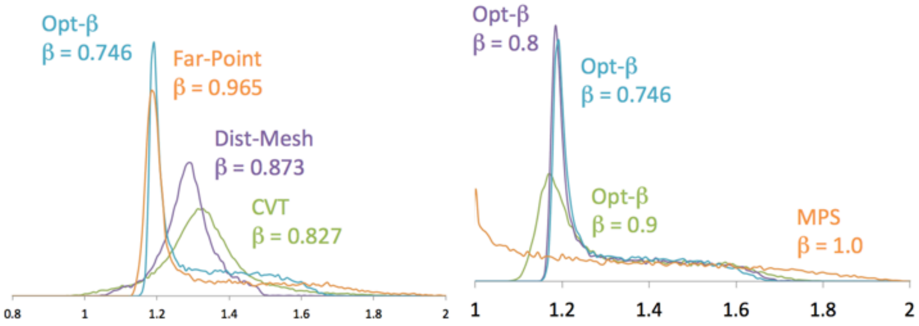


(e) Opt-β^i mesh, $\beta = 0.746$

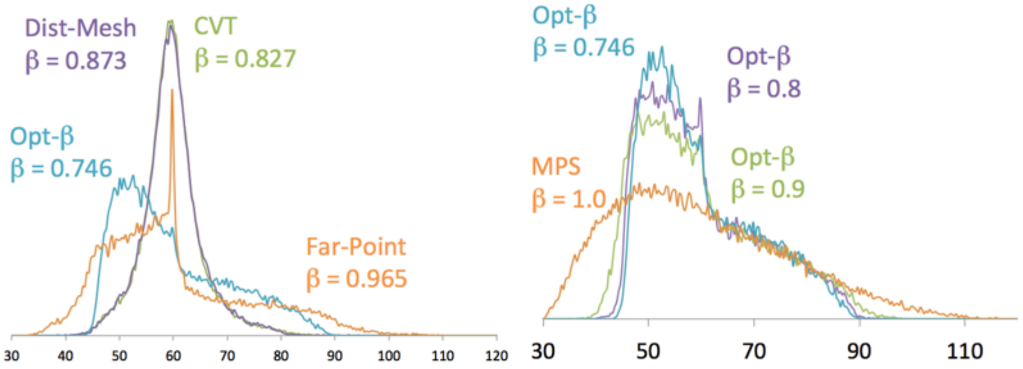
Results: Impact on Mesh Quality



(a) Voronoi Cell Aspect Ratio

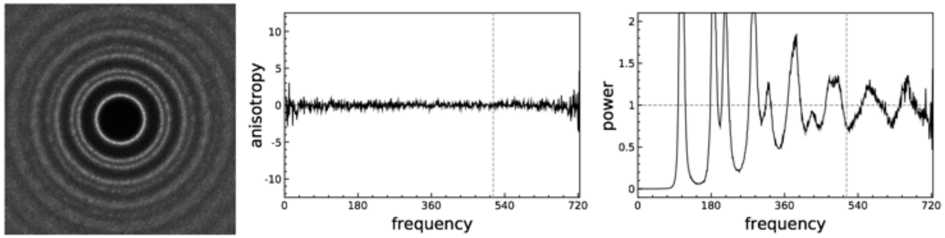


(b) Delaunay Edge Lengths

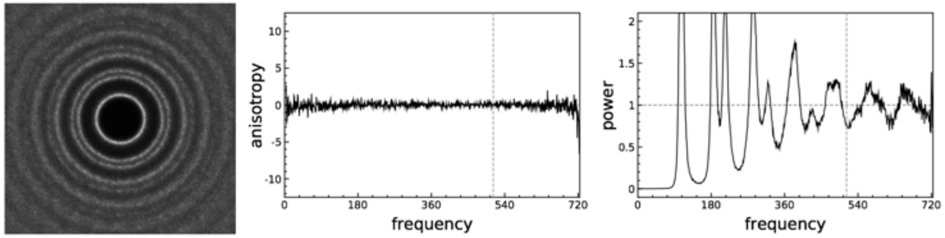


(c) Delaunay Angles

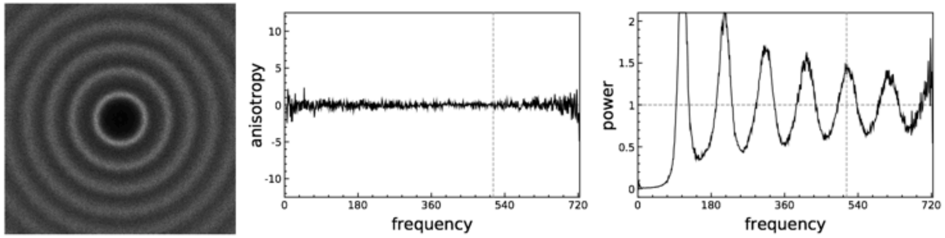
Impact on Noise



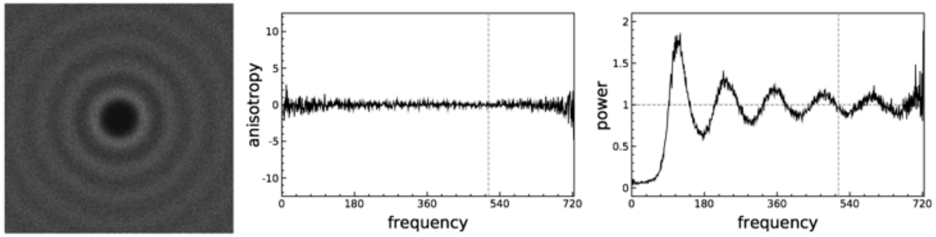
(a) CVT, $\beta = 0.827$



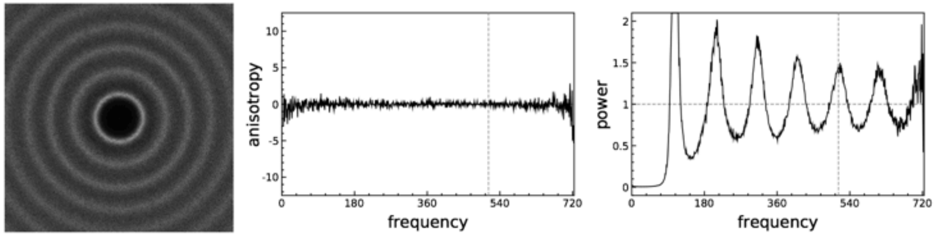
(b) DistMesh, $\beta = 0.873$



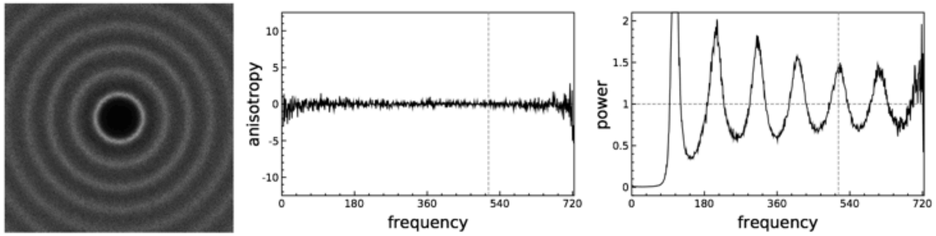
(c) Far-Point, $\beta = 0.965$



(d) MPS, $\beta = 1.0$

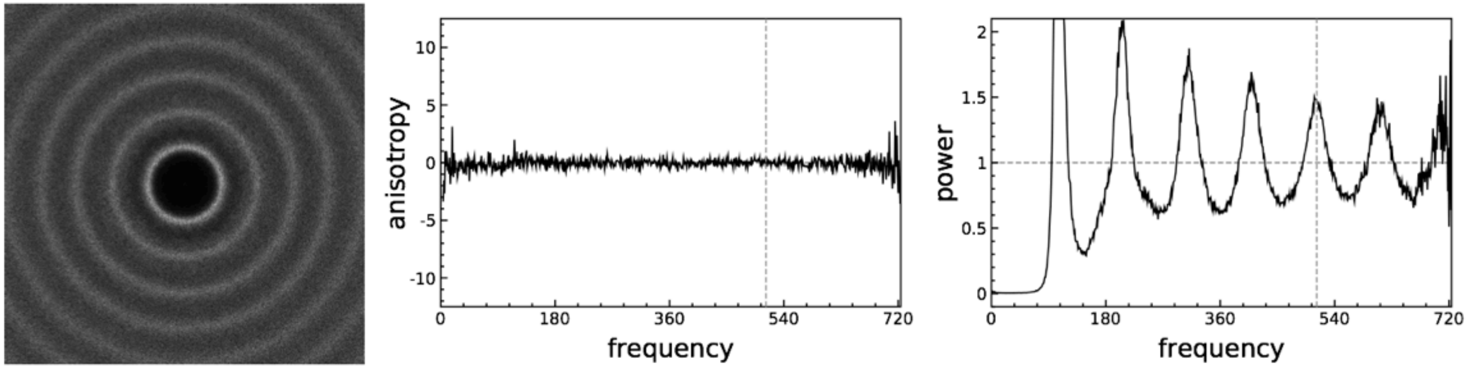


(e) Opt-β^i, $\beta = 0.85$

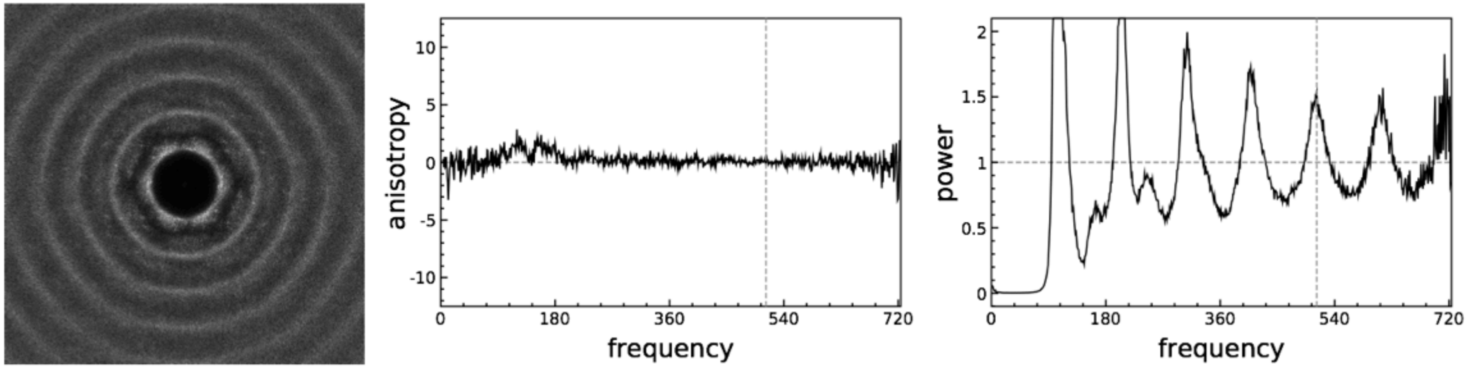


(f) Opt-β^i, $\beta = 0.80$

Impact on Noise

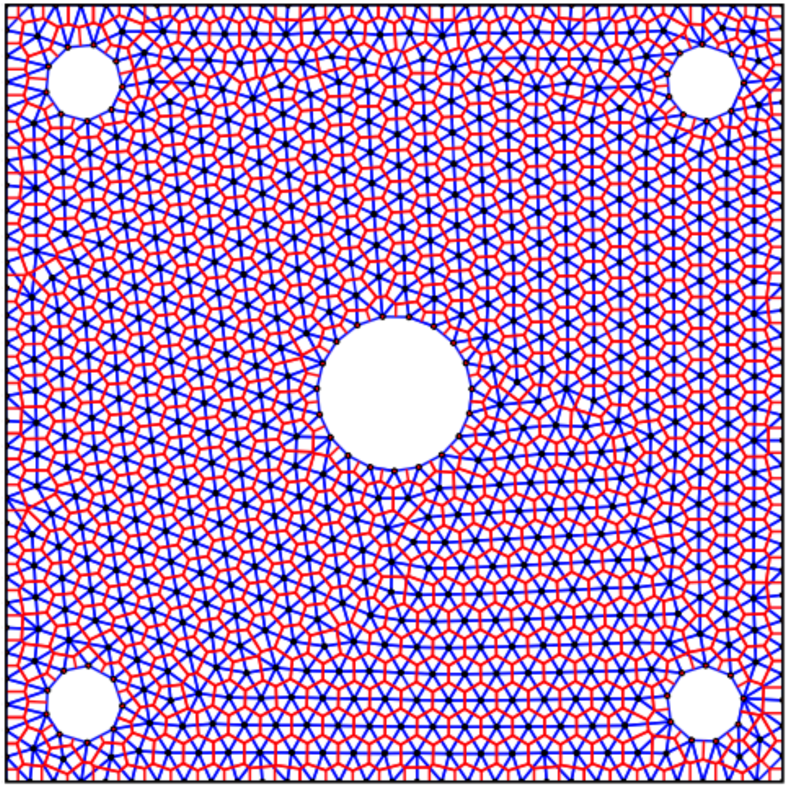


(g) Opt-β^i, $\beta = 0.75$



(h) Opt-β^i, $\beta = 0.70$

Results: Non-convex domains



(e) DistMesh, $\beta = 1.089$

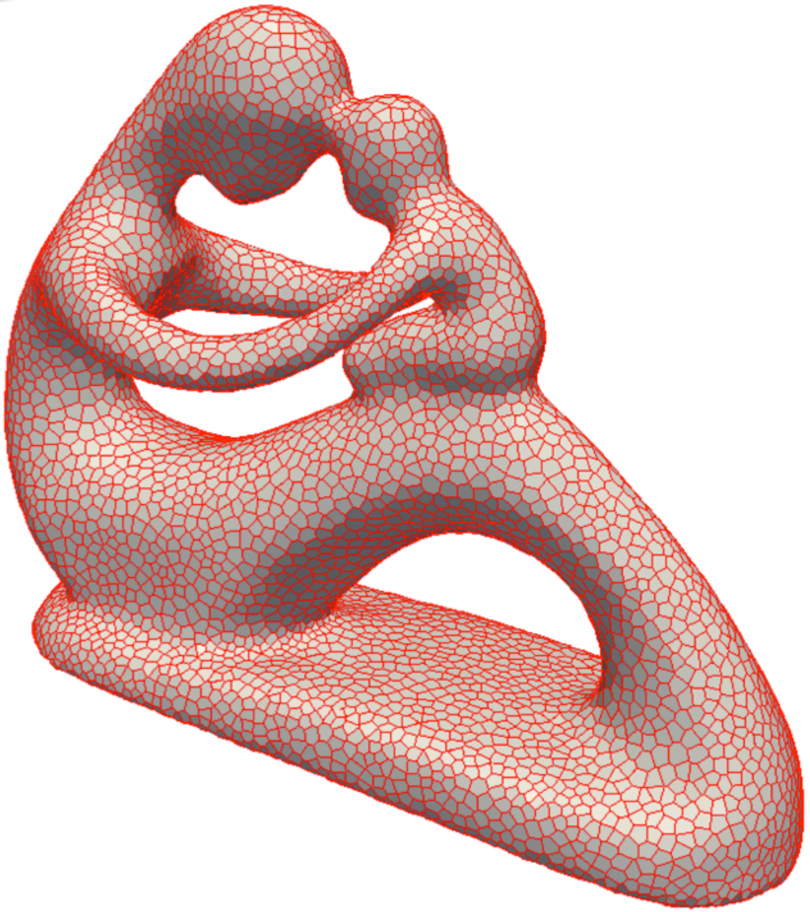
Algorithm	β	$\frac{r_c}{r_{\text{MPS}}}$	$\frac{r_f}{r_{\text{MPS}}}$	min α	max α
MPS	1.0	1.0	1.0	31	115
CVT	1.226	0.931	0.759	24	103
DistMesh	1.089	1.012	0.929	31	114
Far-Point	1.048	1.043	0.996	32	106
Opt-β^i	0.988	0.995	1.007	32	110

(a) Coarse mesh, $r_{\text{MPS}} = 0.0314$, Fig. 10 left column.

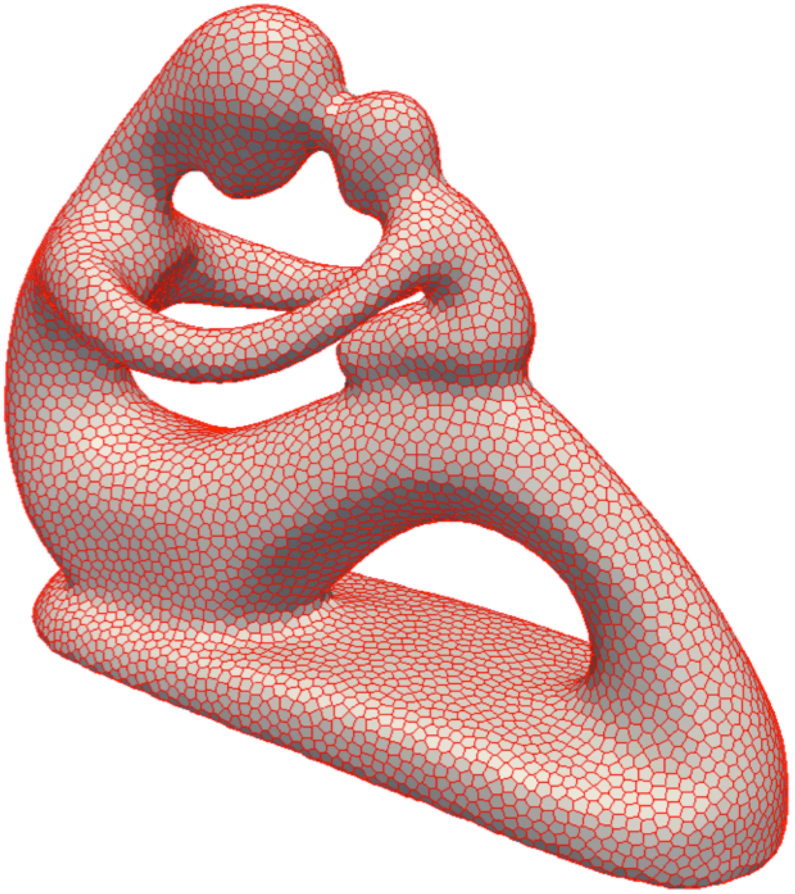
Algorithm	β	$\frac{r_c}{r_{\text{MPS}}}$	$\frac{r_f}{r_{\text{MPS}}}$	min α	max α
MPS	1.0	1.0	1.0	30	117
CVT	1.02	0.989	0.852	33	96
DistMesh	1.07	0.869	0.925	34	107
Far-Point	1.06	1.106	1.047	31	113
Opt-β^i	0.932	0.939	1.008	34	99

(b) Fine mesh, $r_{\text{MPS}} = 0.0157$, Fig. 10 right column.

Results: Curved Surfaces

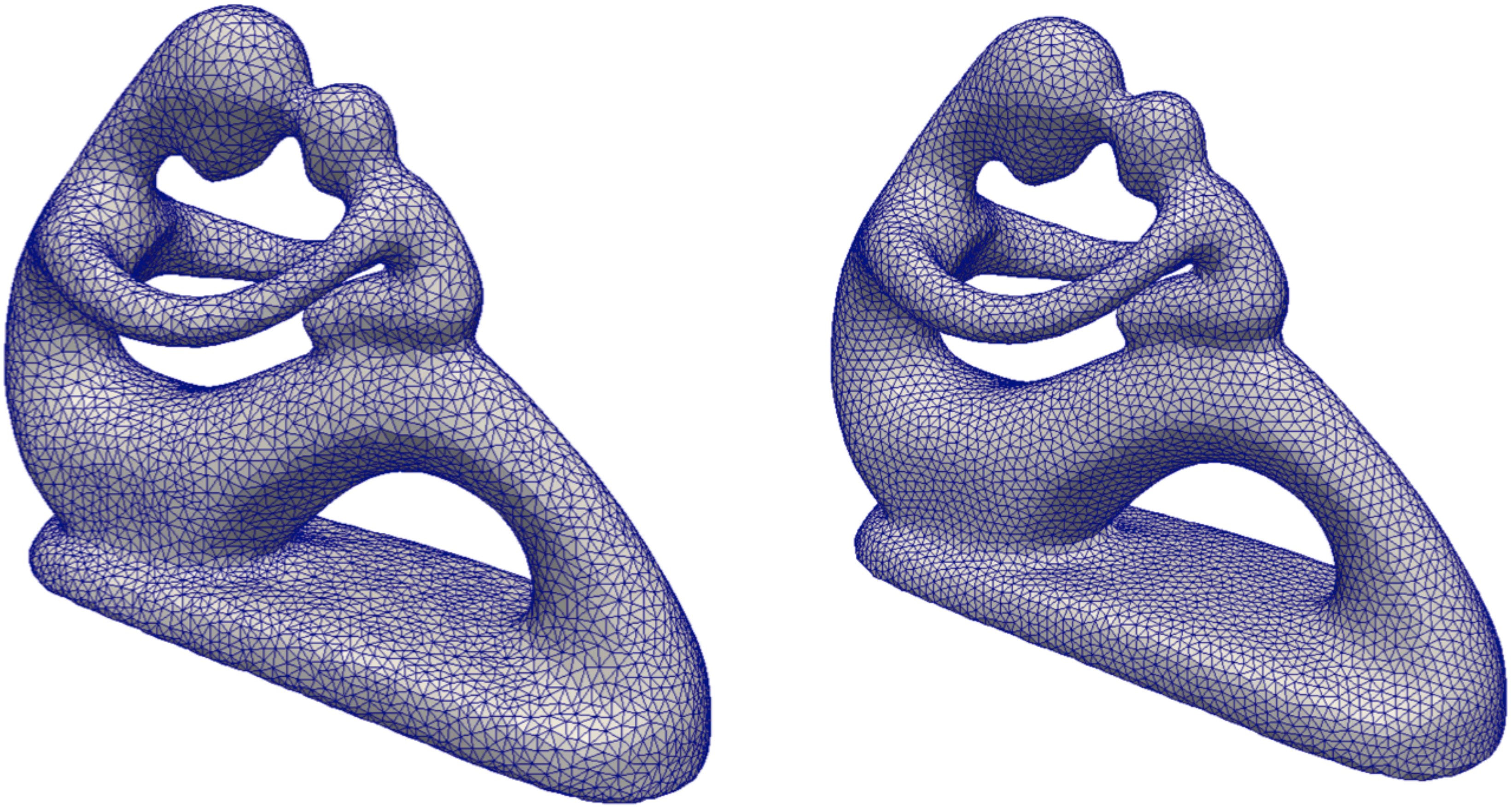


(b) Voronoi input mesh



(d) Voronoi after convergence

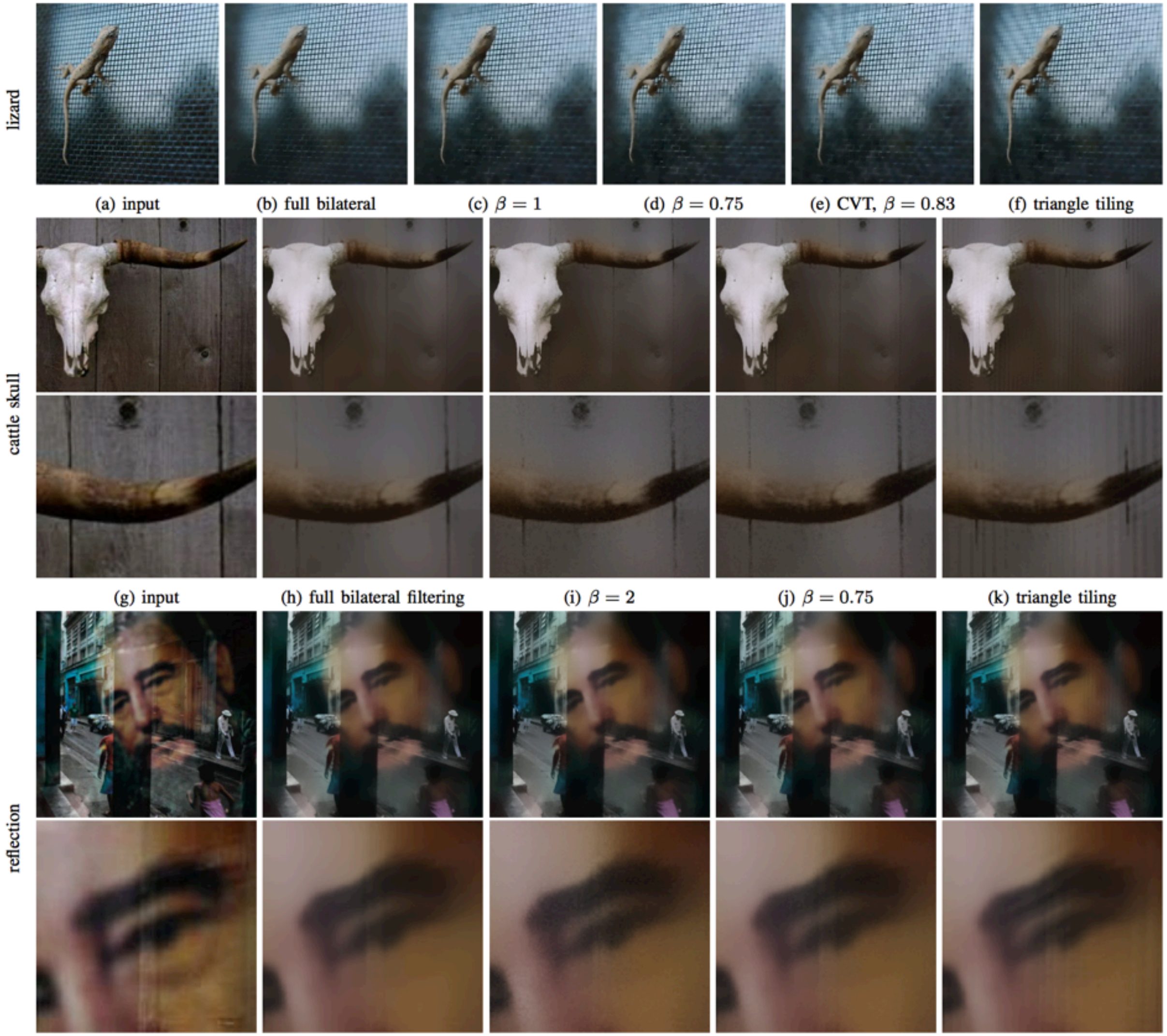
Results: Curved Surfaces



(a) Delaunay input mesh

(c) Delaunay after convergence

Results: Impact on bilateral filtering



(a) input (b) full bilateral (c) $\beta = 1$ (d) $\beta = 0.75$ (e) CVT, $\beta = 0.83$ (f) triangle tiling

(g) input (h) full bilateral filtering (i) $\beta = 2$ (j) $\beta = 0.75$ (k) triangle tiling

(ℓ) input (m) full bilateral filtering (n) $\beta = 2$ (o) $\beta = 0.75$ (p) triangle tiling

Sub-sampling accelerated bilateral filtering. β = 0.75 achieves the right balance between uniformity (reducing noise) and randomness (avoiding aliasing). Notice the noisier results with less uniform sampling (β = 2.0) and more aliasing with more regular sampling (CVT and triangle tiling). For the skull and reflection cases, we show both the whole images and partial zoom-ins.

Summary and Future Work

- This paper introduced a Well-spaced Blue-noise Distribution WBD, with $\beta = r_c/r_f$ measuring coverage uniformity or well-spacedness.

- We proposed the Opt-β algorithm to change a random point set to a WBD; blue noise is preserved up to $\beta \approx 0.75$.

- Extension to higher dimensions esp. impact on slivers is our next step

- Investigate sampling solutions to the Sparse MPS problem.

Thanks! … Questions?