SAND2013-9191C

EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)

Volume 33 (2014), Number 2

Tuning the Discrete Density of a Maximal Random Packing by Disk Injection and Ejection

Eurographics 2014 submission ID paper1102

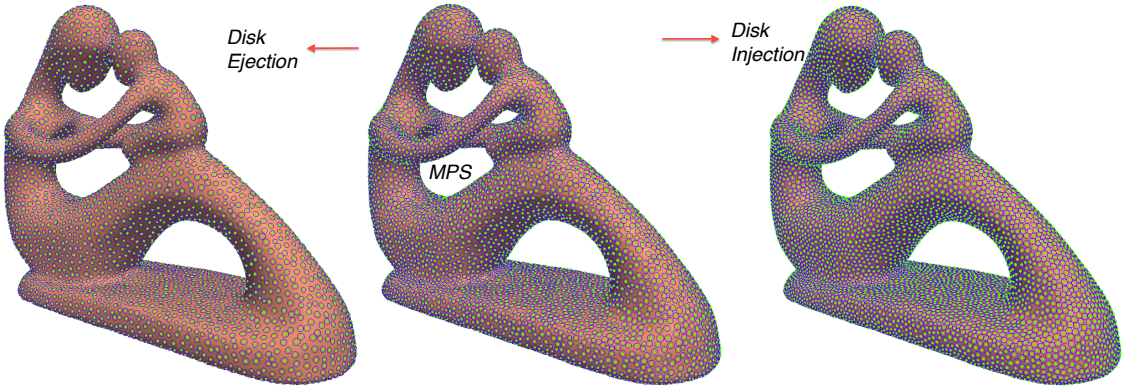


Figure 1: Starting with a Maximal Poisson-disk Sampling (MPS), we may either inject or eject disks to get denser or sparser packings, which are still maximal and satisfy the disk conflict criteria. We can tune the density to (sparse or dense) almost-perfect tilings.

Abstract

We present the Disk Injection and Disk Ejection methods for tailoring the number of disks in a maximal packing to a user-desired density. Both methods move existing disks. Further, in ejection we remove existing disks, and in injection we add new ones. We monotonically change the discrete density of the packing, while maintaining a minimum distance between center points and the maximality of the packing. For injection, we remove a disk, then move it to a corner of the uncovered void left behind. We inject a point if this leaves uncovered space. We follow a series of increasingly aggressive movement rules: movement to a diameter corner, moving all nearby disks away from a central disk, and moving disks towards attracting points. For ejection, we follow similar rules but with opposite movement directions. In both cases, we start with a random maximal Poisson-disk packing, and incrementally move towards a perfect equilateral triangular tiling almost everywhere, with good boundary alignment. The packing retains its good blue noise spectrum for the majority of this range, losing it only as it approaches a structured tiling. We are also able to inject or eject disks to follow a spatially-varying sizing function for the radii, for planar and curved surfaces. We present two applications: modeling fiber reinforced polymers, and Delaunay mesh quality improvement.

1. Introduction

A sphere or disk packing is a set of disks in a domain, such that no disk contains the center of another disk. Another way to define disk packings is to prevent disks from overlapping at all. For constant radius disks, these are equivalent: center-avoiding r disks are equivalent to non-overlapping $r/2$-disks. If the radii vary, then there multiple ways to define a *conflict*, a violation of the center-distance requirement [MREB12]. A maximal packing is one in which the addition of any new disk generates a conflict. This is different than a *maximum* packing; given a maximal packing,

it may be possible to move disks around to make room to add another one without conflict. Disk packings appear frequently in nature, e.g. trees in a forest. In physics, random close packings with fixed locations arise from random sequential adsorption of atoms on a plate. Packings in which points move arise in gasses, liquids, and protein folding.

Generating a triangulation (mesh) of a domain is a prerequisite for many applications. Well-spaced points [Tal97, MTT*96] provide well-shaped meshes. Well-spaced means there is a minimum distance between two sample points, and a maximum distance between a domain point and its nearest sample point, and their ratio is bounded locally. This is equivalent to Voronoi cells having bounded aspect ratio. Well-spacedness is guaranteed by disk packings with uniform or slowly varying disk radii [Tal97, MTT*96]. Given well-spaced points, their Delaunay or weighted Delaunay triangulations provide well-shaped meshes, with many useful properties. The longest edge at a vertex is not much longer than the shortest edge. The discretization error in finite element simulations is bounded by the minimum mesh angle; well-shaped meshes have a lower bound on the minimum triangle angle. The interpolation error is proportional to the aspect ratio of the mesh triangles; this, too, is bounded for meshes of well-spaced points. Standard approaches for generating well-spaced point sets include Delaunay refinement and Poisson-disk sampling.

Delaunay Refinement (DR) [Che89], including its variants such as Voronoi refinement [HMP06], is perhaps the most common technique for generating a triangular mesh. DR incrementally adds points, and maintains a Delaunay triangulation of these points. A triangle with a large empty circumcircle triggers the addition of a new point to destroy it. This can be placed at the center of the circumsphere, nearby, or off-center and closer to the shortest edge [Ü09]. The triangle angle is related by the Central Angle Theorem to both the smallest edge length and the size of the circumsphere. In this way DR produces a sphere packing: the minimum edge length prevents points from being close together, and the maximum circumsphere radius ensures the set is maximal in the sense that there is not enough room to introduce a full-radius disk.

Maximal Poisson-disk Sampling (MPS) is a popular disk packing technique for generating *random* point sets [EPM*11, EMP*12]. Points are added uniformly by area at random, with a minimum separation distance, until the set is maximal. Many methods stop shy of maximality. MPS can generate Voronoi [EM11] and Delaunay meshes [EMD*11], with both constant density and spatially-varying density [MREB12]. As in DR, output points are well-spaced and meshes are well-shaped [MTT*96]. But, because of MPS's randomness, it is preferred over DR for many applications. In computer graphics applications such as texture synthesis, randomness avoids visual artifacts [PH04]. In many sampling and integration applications randomness avoids bias and high variance [SK13]. In fracture simulations, randomness provides realistic crack directions [Bis09, EKL*11]. One potential drawback is that MPS algorithms tend to be slower because they must follow a global random process, whereas DR algorithms are deterministic and local.

Both DR and MPS typically consider point locations to be fixed once a point is generated. For DR, this helps in the termination and output quality proofs. For MPS, this helps in guaranteeing random-looking spectra. However, for many settings this restriction does not make sense; after all, the positions of the points was somewhat arbitrary up to satisfying the conflict criteria. Improving the quality of a given Delaunay triangulation is a clear win over just providing minimum and maximum values arising from disk-packing properties.

In graphics and mesh generation, there are many methods for moving points to improve mesh quality [EAG*13, PS04, SHD11, Knu00, HSD13, DFG99]. Many seek a local optimization of some measure related to mesh quality [Knu00], well-spacedness [EAG*13], or the Fourier spectrum [HSD13], without changing the number of points. Others, such as bubble mesh [SG98] and mesh cleanup [KS07], add and remove points, and explicitly change their connectivity. These are helpful in practice, but it is difficult to show monotonic progress and convergence, or to guarantee the quality of the output.

In the bubble-mesh family of algorithms [SG98], mesh points are moved, removed, and inserted with the goals of achieving good mesh quality and aligning with the domain boundary. Instead of a hard limit on the minimum inter-point distance, points are encouraged to be evenly spaced using spring-like forcing functions.

The physics and statistics communities have extensively studied the density of various types of disk packings [SWM08, Tan79, Pen01], with a different vocabulary and focus. Rectangles, spheres, ellipsoids [DCS*04], irregular shapes, uniform and non-uniform sizes, have all been considered. In physics, most consider dimensions one, two, or three. A disk that cannot move because its tangent disks pin it are "jammed." Adapting disk sets that get closer to maximality with fewer unjammed disks are said to approach the "jamming limit." The distribution of isolated non-jammed "rattler" disks in a maximal packing has been studied. Math and computer science has studied the graph of tangent disks, called the "contact graph" [BR13].

For some physical systems, the disks are packed more (or less) densely than what is typically achieved by uniformly-random sequential insertion. Realism requires more densely packed disks, which may be achieved by movement and insertion. In the physics literature there are many heuristics for moving, injecting, and rejecting disks [RR13]. These are often designed to mimic some physical process, such as atoms moving with a certain velocity and mean-free path as a gas

is compressed; or to predict the density of an unrealized system [DCS*04]. For this paper, the most relevant algorithms are those that try to achieve a density or distribution of some known physical system [NK92].

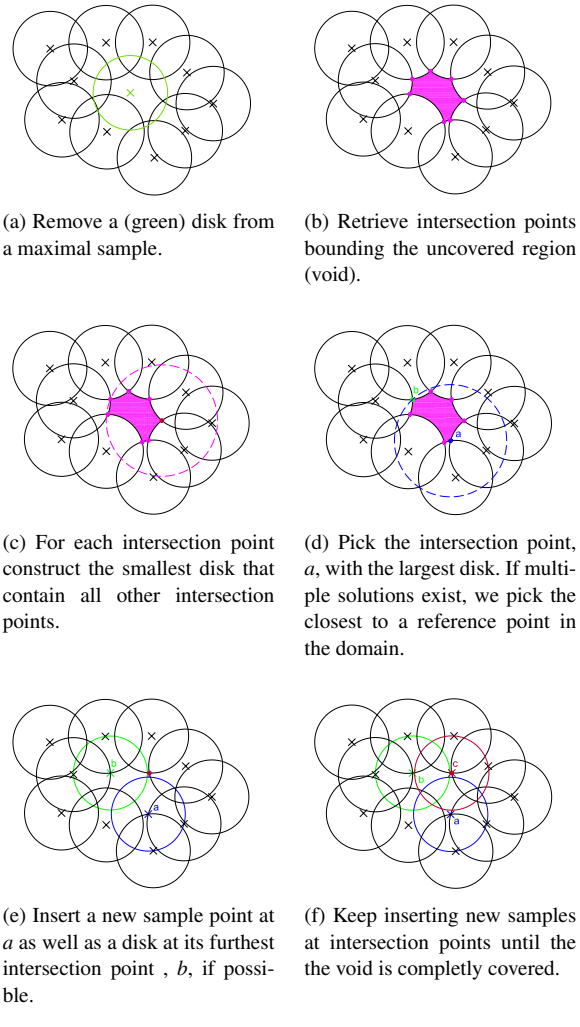
Both disk injection and ejection start with a random input MPS, and keep the disk radii fixed. For injection, we move a disk to some extreme position such that it is as close as possible to other disks, and inject more disks if the remaining uncovered space allows. For ejection, we move a disks' neighbors toward it, then remove it if it is no longer needed for domain coverage. At all steps we avoid conflict and preserve the minimum separation condition. Injection actively reduces the inradius of Voronoi cells, whereas ejection actively increases the outradius of Voronoi cells. In the extreme, both algorithms tend towards equilateral triangle tilings; but ejection has the longest possible edge length that still provides domain coverage, and injection has the shortest possible edge length ensuring point separation. For uniform disk radius (constant sizing function), both generate a large number of equilateral triangles. The majority of the Voronoi cells and Delaunay triangles have small aspect ratios.

Sifted Disks [EMA*13] reduces the number of disks while preserving the packing's maximality: adjacent disks are replaced two-for-one if possible. This is a purely local process that typically gets stuck well before achieving the minimum packing. Disk ejection has the same goal as Sifted Disks, and disk injection the opposite goal. Beyond this superficial contrast, the point movement methods we propose are more aggressive and coordinated, and the output is often close to the maximum (injection) or minimum (ejection) packing.

2. Disk Injection Algorithm

We illustrate the main steps of the disk injection algorithm in Figure 2 using a uniform disk size. We start with a maximal disk packing. We iterate over randomly selected sample points (disk centers). We move this point, and, if its movement created a large enough gap, then we also inject one or more new points. These new sample points preserve the maximal and disk-free conditions, and hence preserve the quality guarantees. We terminate when the user-desired volume fraction is achieved, or no sample point can be relocated.

We have three successively aggressive phases for injecting points. Common to all phases, we conceptually remove a disk and construct the "void" that is left behind. The remaining disks are not maximal. The void is the subdomain that is not covered by a disk; that it, it is the set of locations that a disk center point could be injected without violating the disk-free condition. The void is an arc-gon bounded by a set of circular arcs of the remaining disks. The vertices shared by consecutive arcs are *corners*. Two points are *neighbors* if their disks overlap.



(a) Remove a (green) disk from a maximal sample.

(b) Retrieve intersection points bounding the uncovered region (void).

(c) For each intersection point construct the smallest disk that contain all other intersection points.

(d) Pick the intersection point, a, with the largest disk. If multiple solutions exist, we pick the closest to a reference point in the domain.

(e) Insert a new sample point at a as well as a disk at its furthest intersection point , b, if possible.

(f) Keep inserting new samples at intersection points until the the void is completly covered.

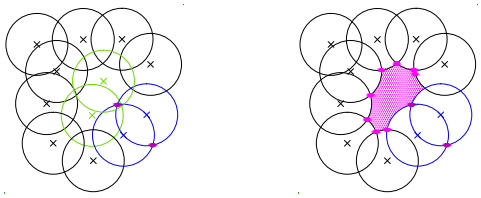
Figure 2: The main steps of Disk-Injection for a uniform sizing function.

2.1. Void Diameter Injection

In the first phase, Void Injection, we move the selected disk center p to one of its void corners. We find the pair of void corners that are farthest apart, the diameter pair a,b. We move p to a. If distance $\mathbf{d}(a,b) < r$, then the entire void is covered, and no other point can be injected. Otherwise, we repeat the process: for the remaining void, we find the diameter pair a_1,b_1, and inject p_1 at a_1, etc.

2.2. Neighbor Repeller Injection

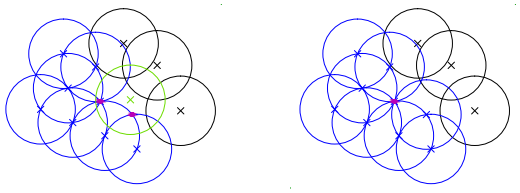
In the second phase, Repeller Injection, we consider a local neighborhood of p. We repel the neighbors of p away from p. Leaving p in place, for each neighbor q, we create its void, and move q to its void corner farthest from p. If this leaves



(a) Multiple disks (green) cover an attractor point (red).

(b) Removing all covering disks leaves the attractor uncovered.

Figure 3: Remove all disks covering an attractor point. Then we can inject a new disk at the attractor. Removal and injection is equivalent to moving one of the original disks.



(a) Remove a (green) disk that is covering two attractor points.

(b) Re-inject it at one of the attractor points.

Figure 4: Injection may create an attractor point that is outside all mobile disks, and so cannot attract a disk.

an uncovered gap we inject a point, just as in Void Injection, and declare success. Otherwise, we move the next neighbor of p. If all neighbors have been moved and no point has been injected, we apply Void Injection on p, moving p to one of its diameter void corners, and injecting if possible.

2.3. Crystal Growth Injection

In the third phase, Crystal Injection, we grow a set of void corner points that are attractors that pull nearby disk centers towards them. We freeze disks as their centers attach to attractor points. This is analogous to crystal growth. We start with two disks with centers at distance r apart; the disk positions are frozen and their intersection points are the first attractor points. We iterate over the mobile (not frozen) disks. We visit these in uniform random order, which is efficient in a parallel implementation. (Other orders, such as visiting those nearby attractors, are possible.)

If a mobile disk does not cover any attractor point, then we move it as in Void Injection.

For a mobile disk $D(p)$ covering attractor point a, we seek to move its disk center p to a. However, a might be strictly inside some other disks, leading to a conflict, so we first delete any such disks; see Figure 3. If moving and/or deleting disks leaves an uncovered void, we immediately inject points to recover maximality.

When we move a disk to an attractor point, the disk is frozen and the attractor point is removed from the attractor set. The disk will intersect with some other frozen disks. These intersection points are added to the pool of attractors, unless they are in strictly inside some already-frozen disk.

See Figure 4 for a case where a disk covers two attractor points, and moving the disk center to one of them leaves the other uncovered. We increase the density further by injecting a disk at these attractors after all disks are frozen. An alternative is to inject there immediately.

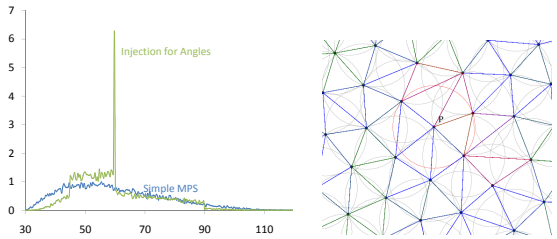


Figure 5: Left, angle distribution after injection for angles. The number of large angles is significantly reduced, but some remain. Right, a stuck case: P is in an obtuse triangle, but none of its neighbors can be moved.

2.4. Injection for Angles.

In the prior algorithms we chose the point to move uniformly at random. There are alternatives that target removing particular structures. In two dimensions, we may target obtuse triangles in a Delaunay triangulation of the points. The circumcircle of an obtuse triangle lies outside it, opposite the obtuse angle. Also, the circumcircle is large compared to the triangle edge length, and the circumcenter v is a Voronoi vertex. Thus an obtuse triangle indicates that the region near v has no nearby sample points. Moving sample points even farther away from v, using v as a repeller in Repeller Injection, is likely to allow the injection of a new sample point at v or nearby. See Figure 5 for typical results.

3. Disk Ejection Algorithms

We may modify Disk Injection from Section 2 to remove disks. The analog to Void Injection has already been done, and is Sifted Disks [MREB12], replacing overlapping disks two-for-one if possible.

3.1. Neighbor Attractor Ejection

Attractor Ejection is a modification of Repeller Injection from Section 2.2, but moving disks in the opposite direction. Take an arbitrary disk center p, and consider each neighboring disk center q in sequence. Move q to q', the point

on segment $\overline{pq}$ as close as possible to p while still covering its void, and keeping it outside all disks except perhaps p's. (The next paragraph explains how to calculate q'.) That is, moving q will not uncover some part of the domain. After all neighbor movements, remove p if its new void is already covered by its neighbors. Otherwise, we find a new position for p. Move p to the center of the minimum disk covering its void's corners. If the new position is inside some disk (the void is not convex) then project it to the void. In rare cases there may be no new position for p that covers its void; in that case we put p and its neighbors back in their original positions.

To find q', we compute q's void corners. For each corner we find the interval of $\overline{pq}$ within r of it. We intersect all these intervals. The endpoint of the remaining interval closest to p is q'.

3.2. Crystal Growth Ejection

Find (or create if necessary) two disks with center distance $\sqrt{3}r$. Dilate their disks to $\sqrt{3}r$; their two intersection points are attractor points, where a new disk would create equilateral triangles of the maximum possible edge length, while still having the center of the triangle covered by r-disks. Pick any other sample q at random. If its disk does not cover an attractor point, move it as in Attractor Ejection. Otherwise, move q to the (closest) attractor point that its disk covers. If this creates a conflict, remove the other conflicted disks (not q) and resample any voids to regain maximality. Then update the attractor points.

4. Visiting the Phases and Achieved Area Fraction

The user specifies the exact number of points desired, N. Since the domain and disk radius are fixed, we can calculate this from a user specified relative radius ρ, area fraction or packing density η [EAG*13].

We apply the phases in sequence. If at any time the current number of points n reaches N, we are done. Figure 6 shows the typical progress if one phase is continued ad infinitum. Note that each has rapid progress initially, then minor progress after a certain threshold. We use these thresholds as the decision criteria for transitioning from one phase to the next. In particular, we transition from Void Injection to Repeller Injection when the area fraction reaches 0.75, and from Repeller Injection to Crystal Injection when it reaches 0.79. Note Crystal Injection can typically reach about 0.89 in a periodic domain, after which progress is slow. The densest hex packing of an infinite domain has $\pi/(2\sqrt{3}) \approx 0.907$; periodic domains have a smaller limit because of boundary effects, the interaction between the periodic distance and r.

5. Results for a Periodic Unit Box

We apply our methods to the unit box with periodic boundary conditions and measure the output. Figure 7 shows the

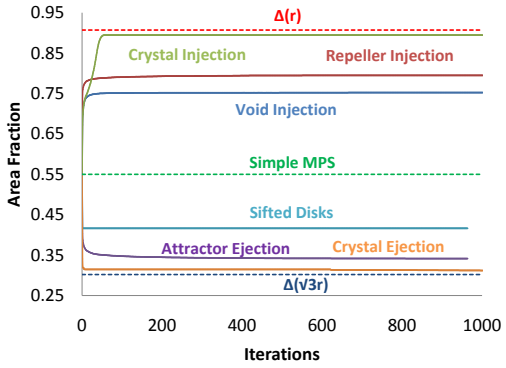


Figure 6: Disk injection and ejection can achieve a wide range of area fractions. MPS provides the input, at around 0.55. Ejection can tune this down to 0.31, and injection can tune it up to 0.89. The densest hex packing has $\pi/(2\sqrt{3}) \approx 0.907$ and the sparsest has $\pi/(6\sqrt{3}) \approx 0.302$.

effect of changing the area fraction over the different quality measures. Spectral analysis for various area fractions and Simple MPS are shown in Figure 9.

6. Algorithms for Spatially Varying Radii

We extend our methods to non-uniform disk radii, that varies across the domain. In some cases the only differences are that the number of neighbors may be larger, and we need to consider power diagrams and power vertices instead of Voronoi diagrams and Voronoi vertices [YW13, EMA*13] when finding void corners.

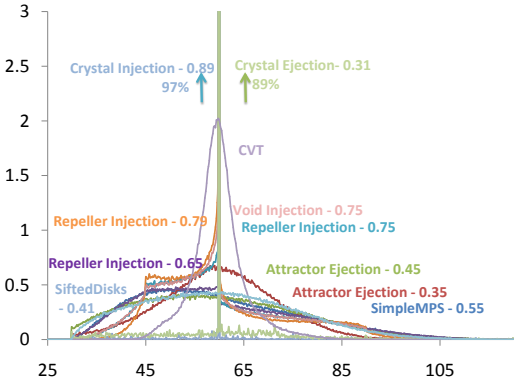
We define the *weighted distance* $\mathbf{w}(a,b)$ from a to b to be $\mathbf{d}(a,b) - r(a)$, as in additively weighted Voronoi diagrams. Note the distance is asymmetric, $\mathbf{w}(a,b) \neq \mathbf{w}(b,a)$.

A key choice is how to define conflict. We choose a variant related to the **prior-disk** and **smaller-disk** criteria [MREB12]. In particular, when we move or inject a disk we consider it to be the latest arrival, and allow its center to be placed anywhere not already covered by the other disks. That is, we accept any disk center if the weighted distances of the other centers to it are all positive. It is possible for the new disk to be large enough that it covers the center of some other (prior) disk. This changes the arrival order of the disks, but, as in **smaller-disk**, it ensures that no pair of disks cover each others' centers.

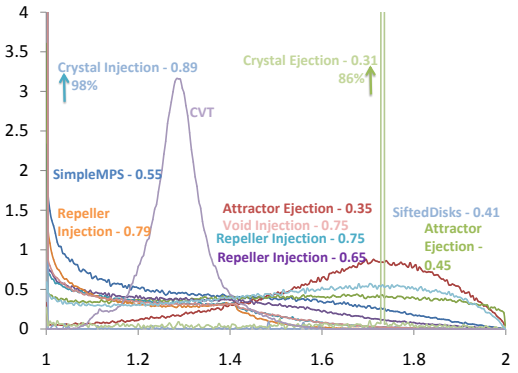
6.1. Non-uniform Injection

6.1.1. Void Injection

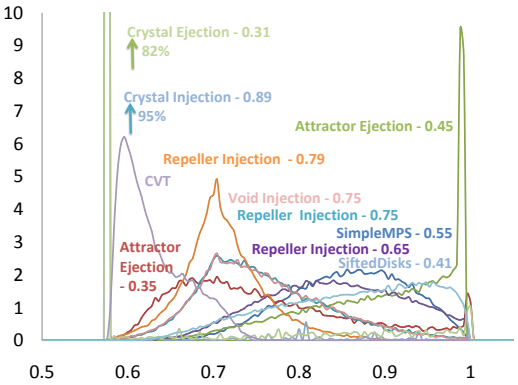
We consider removing one disk and replacing it by two (or more). However, the disk radius is potentially different at each void corner. For each corner c, its *weighted diameter* is its maximum weighted distance to another corner,



(a) Delaunay Angles



(b) Delaunay Edge Lengths



(c) Local Beta

Figure 7: Quality achieved by different algorithms. From an initial 100K sample, we can tune its area fraction to any value in $[0.31, 0.79]$. Comparing between the different injection and ejection algorithms, to the input Simple MPS, and Centroidal Voronoi Tessellation (CVT)

$\max_i \mathbf{w}(c, c_i)$. We remove p, and replace it with a disk at the corner with the maximum weighted diameter. If the diameter is negative, then this disk covers all other corners and injection failed. Otherwise, there are some uncovered corners, and we recursively compute the new void(s) and inject the new corner with the maximum weighted diameter. We stop when maximality is recovered.

6.1.2. Repeller Injection

Given disk p, we consider moving each of its neighbors q in turn. We move q to its void corner c with maximum $\mathbf{w}(c, p)$. If this leaves a void, then we recursively insert a new disk at c' with maximum $\mathbf{w}(c', p)$.

There is no Crystal Injection or ejection for non-uniform radii.

6.2. Non-uniform Ejection

For Sifted Disks, we use Sifted Disks [EMA*13]. For Attractor Ejection, we start by assuming that the disk radius at q is invariant, and estimate q's new position q' exactly as before. The problem is that its new radius might be too small to cover its original void. This would destroy maximality and require *adding* points. So, if that happens, we do a simple heuristic search for an acceptable position. We try $q'' = q + 0.9\vec{qq'}$. We try this scaling up to three times total, stopping if the original void is covered.

While this works reasonably well in practice, many other strategies for searching for good positions are possible, such as moving q' farther if its covers the void by a wide margin, or moving q' to the corner closest to p.

7. Meshing Applications

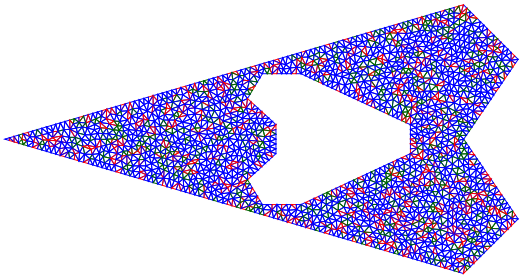
In this section we demonstrate how our method improves mesh quality of non-convex domains, and curved surfaces. For domains with boundaries, we place disks with centers on the one-dimensional domain boundary. We first adjust the spacing of these disks using one-dimensional disk injection or ejection, respecting the sizing function, and keeping disk centers on the boundary curves. These disks are fixed and not moved when processing the rest of the surface.

7.1. Non-convex domains

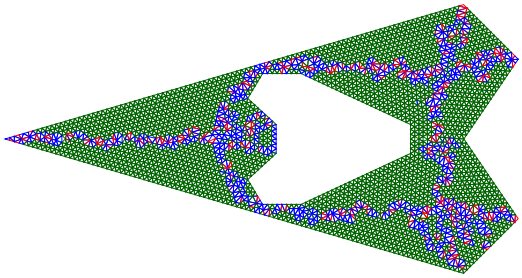
We applied Disk Injection over a non-convex domain with a hole. Figure 8 shows the Delaunay mesh for the input Simple MPS, and after injection.

7.2. Curved surfaces

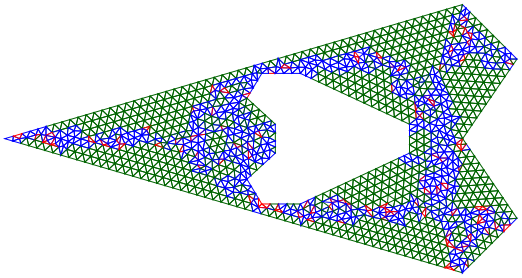
The algorithmic changes are the following. We use three-dimensional disks. The distance between centers is based on Euclidean distance between the spheres, which will be close



(a) Simple MPS with boundary samples.

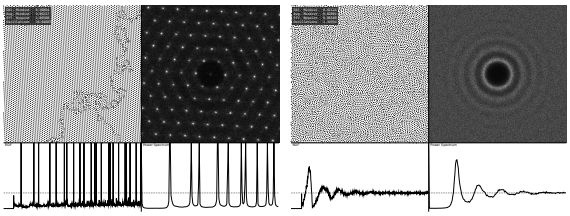


(b) Crystal Growth Injection with boundaries.

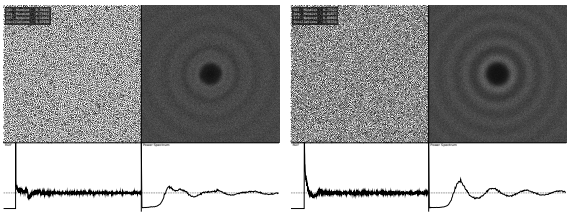


(c) Crystal Growth Ejection with boundaries.

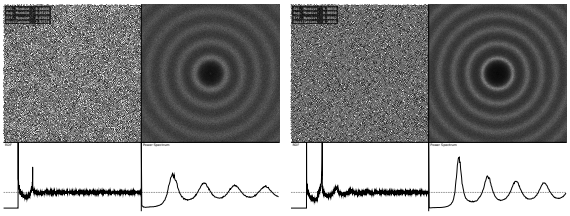
Figure 8: Disk Injection over a bounded non-convex domain with a hole.



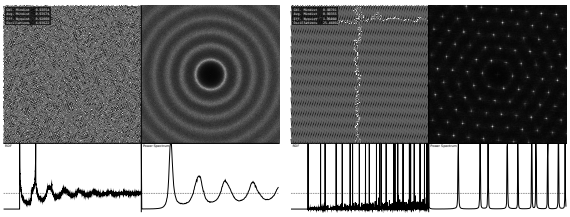
(a) Area fraction = 0.31 (b) Area fraction = 0.35



(c) Area fraction = 0.45 (d) Area fraction = 0.55 (Simple MPS)



(e) Area fraction = 0.65 (f) Area fraction = 0.75



(g) Area fraction = 0.8 (h) Area fraction = 0.89

Figure 9: Spectral analysis for Simple MPS, Disk Ejection, and Disk Injection for various area fractions.

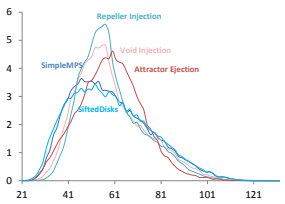
to geodesic distance provided that the surface curvature is small or it is sampled fine enough. We define voids using the complement of the intersection of spheres with the surface. That is, voids lie on the surface, despite the use of spheres. Thus injected points lie on the surface.

7.2.1. Fertility Sculpture Sampling

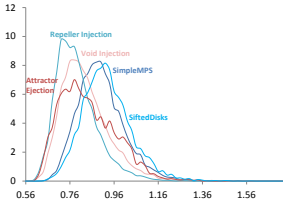
We sampled the well-known Fertility model, sizing disk radii based on the local curvature. Then we applied Disk Ejection and Disk Injection using the same sizing function. See Figure 10 for results.

8. Fiber Composite Models

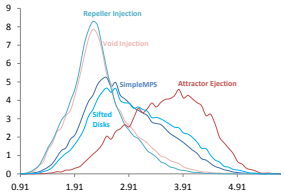
We seek to model the micro-structure of a unidirectional E-glass fiber reinforced epoxy material. The composite consists of an arrangement of fibers embedded in a matrix material. For a unidirectional material, the fibers are aligned in roughly the same direction. Since the fibers have circular cross sections of about the same diameter, the material cross section looks like a random two-dimensional arrangement of non-overlapping unit disks. The extent of the material is usually orders of magnitude larger than the fiber diameter, so the material is modeled well by a periodic arrangement of disks in a square, called a "representative volume element." Many have tried to characterize material strength and stress response using a structured (hexagonal, square, ...) fiber packing [LBM00, BAC05, SOK01, Mal08, TTL06]. While these structured models produce reasonable results



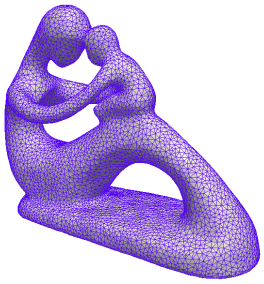
(a) Delaunay Angles



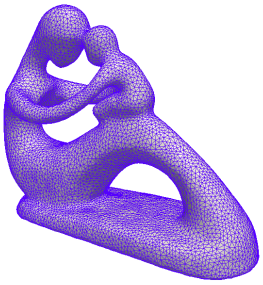
(b) Local Beta



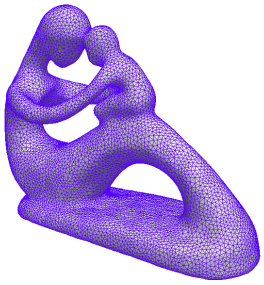
(c) Delaunay Edge Lengths



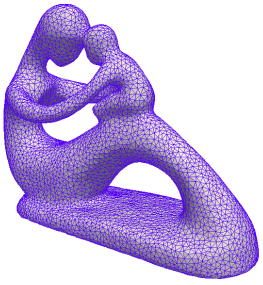
(d) Simple MPS



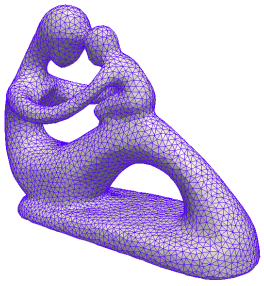
(e) Void Injection



(f) Repeller Injection

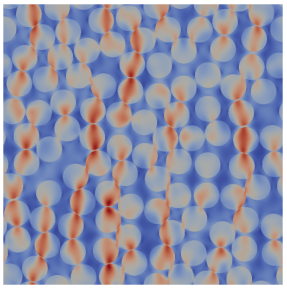


(g) Sifted Disks

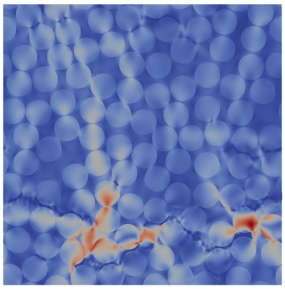


(h) Attractor Ejection

Figure 10: Delaunay meshes and quality measures of Disk Injection and Disk Ejection on the Fertility curved surface model. We were able to reduce the mesh size by 38% and have a better quality than MPS using Attractor Ejection. Repeller Injection gives even better quality and increases the mesh size by 44%



(a) peak load



(b) post fracture

Figure 11: Von Mises stress distribution under uniaxial transverse tension.

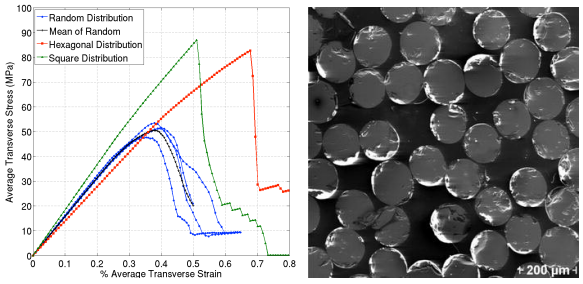
under elastic loading, they significantly over-predict strength and damage resistance. To accurately predict crack initiation and propagation in a homogenized multi-scale analysis, a random packing is necessary [WCS98, TGL08, Rom10, SGP06, SG06, TTL06]. For good fidelity, the density of the disks in the packing must also match the density of the fibers in the material. Density is typically measured by volume fraction, the fraction of the square domain covered by some disk. MPS and its near-maximal variants usually produce volume fractions around 50%. In physical materials, the volume fraction is often larger, 60–70%; see Figure 12b. (In addition, dart throwing does not represent the physical process of packing fibers.)

8.1. Our Fiber Simulations

We start with an MPS over a periodic unit square, a representative volume element. We use disk injection to increase the volume fraction to the desired volume fraction matching the material at hand, in our case 67%. We load the volume element transverse to the fibers; see Figure 11. We calculate the boundary conditions using a multi-scale approach, solving for the relative velocities at the periodic nodes, ensuring the homogenized response is uniaxial. We ran four examples, each with different random MPS input and injected disks, an averaged them. Figure 12 shows the four simulated responses, plus their average. Note that the simulations span a small range in the elastic range, but at fracture (peak load) there is a wider variation in the responses. This is consistent with experiments. We also ran simulations with structured packings; we can see from Figure 12 that the responses are unrealistic.

Conclusions

We have introduced a method to tune the discrete density of a (random) maximal disk packing to any feasible user-specified value. In contrast to prior methods, we are able to get much closer to the densest-possible and sparsest-possible



(a) simulated responses (b) physical fiber micrograph

Figure 12: (a) Tensile response at $v_f = 67\%$. Multiple random-sample simulations (blue), modeling natural material variations, and their average (black); and unrealistic structured-mesh responses (red and green). (b) Scanning electron microscope micrograph of a fiber composite cross section [BRV*08].

packings. Disks are added or removed one by one, giving very fine control. Blue noise is retained for much of this range, but lost as we approach a structured tiling.

We have demonstrated the efficiency of our method using uniform sizing functions over planar domains, and curved domains typical of graphics models. We show the usefulness of our output for matching the density of physical materials, and generating realistic fracture simulations.

For future work, we would like to consider methods to reintroduce randomness. Injected disks are exactly the disk radius away from neighbors. Ejected disks tend to be equidistant to several nearby disks. Both of these might be perturbed. We have demonstrated that the method works for a slowly varying sizing function; we would like to explore the limits of how fast the distribution may be graded.

References

[BAC05] BARBERO E., ABDELAL G., CACERES A.: A micromechanics approach for damage modeling of polymer matrix composites. *Composite Structures 67*, 4 (2005), 427 – 436. doi:10.1016/j.compstruct.2004.02.001. 7

[Bis09] BISHOP J.: Simulating the pervasive fracture of materials and structures using randomly close packed Voronoi tessellations. *Comput. Mech. 44* (2009), 455–471. doi:10.1007/s00466-009-0383-6. 2

[BR13] BEZDEK K., REID S.: Contact graphs of unit sphere packings revisited. *Journal of Geometry 104*, 1 (2013), 57–83. doi:10.1007/s00022-013-0156-4. 2

[BRV*08] BRAUER D., RÜSSEL C., VOGT S., WEISSER J., SCHNABELRAUCH M.: Degradable phosphate glass fiber reinforced polymer matrices: mechanical properties and cell response. *J of Materials Science: Materials in Medicine 19*, 1 (2008), 121–127. doi:10.1007/s10856-007-3147-x. 9

[Che89] CHEW L. P.: *Guaranteed-Quality Triangular Meshes*. Tech. Rep. 89-983, Department of Computer Science, Cornell University, 1989. 2

[DCS*04] DONEV A., CISSE I., SACHS D., VARIANO E. A., STILLINGER F. H., CONNELLY R., TORQUATO S., CHAIKIN P. M.: Improving the density of jammed disordered packings using ellipsoids. *Science 303*, 5660 (2004), 990–993. doi:10.1126/science.1093010. 2, 3

[DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal Voronoi tessellations: Applications and algorithms. *SIAM review 41*, 4 (1999), 637–676. 2

[EAG*13] EBEIDA M. S., AWAD M. A., GE X., MAHMOUD A. H., MITCHELL S. A., KNUPP P. M., WEI L.-Y.: Improving spatial coverage while preserving blue noise of point sets. *Computer-Aided Design 0*, 0 (2013), –. In Press, Accepted Manuscript. doi:10.1016/j.cad.2013.08.015. 2, 5

[EKL*11] EBEIDA M. S., KNUPP P. M., LEUNG V. J., BISHOP J. E., MARTINEZ M. J.: Mesh generation for modeling and simulation of carbon sequestration process. In *DOE Scientific Discovery through Advanced Computing (SciDAC)* (July 2011). 2

[EM11] EBEIDA M. S., MITCHELL S. A.: Uniform random Voronoi meshes. In *Int. Meshing Roundtable* (2011), pp. 258–275. 2

[EMA*13] EBEIDA M. S., MAHMOUD A. H., AWAD M. A., MOHAMMED M. A., MITCHELL S. A., RAND A., OWENS J. D.: Sifted disks. *Computer Graphics Forum 32*, 2pt4 (2013), 509–518. 3, 5, 6

[EMD*11] EBEIDA M. S., MITCHELL S. A., DAVIDSON A. A., PATNEY A., KNUPP P. M., OWENS J. D.: Efficient and good Delaunay meshes from random points. *Comput. Aided Des. 43*, 11 (2011), 1506–1515. doi:10.1016/j.cad.2011.08.012. 2

[EMP*12] EBEIDA M. S., MITCHELL S. A., PATNEY A., DAVIDSON A. A., OWENS J. D.: A simple algorithm for maximal Poisson-disk sampling in high dimensions. *Computer Graphics Forum 31*, 2 (May 2012), 785–794. doi:10.1111/j.1467-8659.2012.03059.x. 2

[EPM*11] EBEIDA M. S., PATNEY A., MITCHELL S. A., DAVIDSON A., KNUPP P. M., OWENS J. D.: Efficient maximal Poisson-disk sampling. *ACM Trans. Graphics 30*, 4 (2011), 49:1–49:12. doi:10.1145/1964921.1964944. 2

[HMP06] HUDSON B., MILLER G., PHILLIPS T.: Sparse Voronoi refinement. In *Int. Meshing Roundtable* (2006), Sandia National Laboratories, pp. 339–358. 2

[HSD13] HECK D., SCHLÖMER T., DEUSSEN O.: Blue noise sampling with controlled aliasing. *ACM Trans. Graphics 32*, 3 (July 2013), 25:1–25:12. doi:10.1145/2487228.2487233. 2

[Knu00] KNUPP P. M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II—A framework for volume mesh optimization and the condition number of the Jacobian matrix. *Int. J. for Numerical Methods in Engineering 48*, 8 (2000), 1165–1185. 2

[KS07] KLINGNER B. M., SHEWCHUK J. R.: Aggressive tetrahedral mesh improvement. In *Int. Meshing Roundtable* (2007), pp. 3–23. 2

[LBM00] LANDIS C. M., BEYERLEIN I. J., MCMEEKING R. M.: Micromechanical simulation of the failure of fiber reinforced composites. *J Mechanics and Physics of Solids 48*, 3 (2000), 621 – 648. doi:10.1016/S0022-5096(99)00051-4. 7

[Mal08] MALIGNO A. R.: *Finite element investigations on the microstructure of composite materials*. PhD thesis, University of Nottingham, 2008. 7

[MREB12] MITCHELL S. A., RAND A., EBEIDA M. S., BAJAJ C.: Variable radii Poisson-disk sampling, extended version. In *Canadian Conference on Computational Geometry* (2012), pp. 1–9. 1, 2, 4, 5

[MTT*96] MILLER G. L., TALMOR D., TENG S.-H., WALKINGTON N., WANG H.: Control volume meshes using sphere packing: Generation, refinement and coarsening. In *Int. Meshing Roundtable* (1996), pp. 47–61. 2

[NK92] NOLAN G., KAVANAGH P.: Computer simulation of random packing of hard spheres. *Powder Technology 72*, 2 (1992), 149 – 155. doi:10.1016/0032-5910(92)88021-9. 3

[Pen01] PENROSE M. D.: Random parking, sequential adsorption, and the jamming limit. *Communications in Mathematical Physics 218* (2001), 153–176. doi:10.1007/s002200100387. 2

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. 2

[PS04] PERSSON P.-O., STRANG G.: A simple mesh generator in MATLAB. *SIAM review 46*, 2 (2004), 329–345. 2

[Rom10] ROMANOWICZ M.: Progressive failure analysis of unidirectional fiber-reinforced polymers with inhomogeneous interphase and randomly distributed fibers under transverse tensile loading. *Composites Part A: Applied Science and Manufacturing 41*, 12 (2010), 1829–1838. 8

[RR13] RIMOLI J. J., ROJAS J. J.: Meshing strategies for the alleviation of mesh-induced effects in cohesive element models. *Computational Physics* (2013). arXiv:1302.1161. 2

[SG98] SHIMADA K., GOSSARD D.: Automatic triangular mesh generation of trimmed parametric surfaces for finite element analysis. *Computer Aided Geometric Design 15*, 3 (1998), 199–222. 2

[SG06] SWAMINATHAN S., GHOSH S.: Statistically equivalent representative volume elements for unidirectional composite microstructures: Part II—With interfacial debonding. *J Composite Materials 40*, 7 (2006), 605–621. 8

[SGP06] SWAMINATHAN S., GHOSH S., PAGANO N.: Statistically equivalent representative volume elements for unidirectional composite microstructures: Part i-without damage. *J Composite Materials 40*, 7 (2006), 583–604. 8

[SHD11] SCHLÖMER T., HECK D., DEUSSEN O.: Farthest-point optimized point sets with maximized minimum distance. In *HPG '11* (2011), pp. 135–142. doi:10.1145/2018323.2018345. 2

[SK13] SUBR K., KAUTZ J.: Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration. *ACM Trans. Graphics 32*, 4 (July 2013), 128:1–128:12. doi:10.1145/2461912.2462013. 2

[SOK01] SEARLES K., ODEGARD G., KUMOSA M.: Micro- and mesomechanics of 8-harness satin woven fabric composites: I — evaluation of elastic behavior. *Composites Part A: Applied Science and Manufacturing 32*, 11 (2001), 1627 – 1655. doi:10.1016/S1359-835X(00)00181-0. 7

[SWM08] SONG C., WANG P., MAKSE H. A.: A phase diagram for jammed matter. *Nature 453* (2008), 629–632. doi:10.1038/nature06981. 2

[Tal97] TALMOR D.: *Well-Spaced Points for Numerical Methods*. PhD thesis, Carnegie Mellon University, Pittsburgh, August 1997. CMU CS Tech Report CMU-CS-97-164. 2

[Tan79] TANEMURA M.: On random complete packing by discs. *Annals of the Institute of Statistical Mathematics 31* (1979), 351–365. doi:10.1007/BF02480293. 2

[TGL08] TOTRY E., GONZÁLEZ C., LLORCA J.: Failure locus of fiber-reinforced composites under transverse compression and out-of-plane shear. *Composites Science and Technology 68*, 3 (2008), 829–839. 8

[TTL06] TAY T.-E., TAN V. B., LIU G.: A new integrated micro–macro approach to damage and fracture of composites. *Materials Science and Engr. B 132*, 1 (2006), 138–142. 7, 8

[Ü09] ÜNGÖR A.: Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. *Computational Geometry: Theory and Applications 42* (2009), 109–118. doi:10.1016/j.comgeo.2008.06.002. 2

[WCS98] WERWER M., CORNEC A., SCHWALBE K.-H.: Local strain fields and global plastic response of continuous fiber reinforced metal-matrix composites under transverse loading. *Computational materials science 12*, 2 (1998), 124–136. 8

[YW13] YAN D.-M., WONKA P.: Gap processing for adaptive maximal Poisson-disk sampling. *ACM Trans. Graphics 28*, 4 (2013), in press. SIGGRAPH 2013. URL: http://peterwonka.net/Publications/pdfs/2013.TOG.Dongming.MaximalPoissonSampling.pdf. 5