Feature-preserving direct blue noise sampling for surface meshes

Jean-Luc Peyrot[1], Frédéric Payan[1] and Marc Antonini[1]

[1]Laboratory I3S - UMR 7271 - CNRS, University of Nice - Sophia Antipolis (France)

Abstract

We present a new direct Poisson disk sampling for surface meshes. Our objective is to sample triangular meshes, while satisfying good blue noise properties, but also preserving features. Our method combines a feature detection technique based on vertex curvature, and geodesic-based dart throwing. Our method is fast, automatic, and experimental results prove that our method is well-suited to CAD models, since it handles sharp features and high genus meshes, while having good blue noise properties.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Sampling

1. Introduction

Sampling is a challenging task for any numerical data processing, in order to avoid artefacts due to aliasing problems [Cro77].

Nowadays, Poisson disk distribution is widely used in computer graphics, and considered as one of the best sampling patterns for many applications. The Poisson disk sampling of a surface ensures a uniform point distribution over the surface while having spatial irregularities. This uniform but irregular distribution, most of times generated randomly, satisfies the so-called blue noise properties, and overcomes the aliasing problems. In 2D cases, blue noise properties are satisfactory when the resulting 2D periodogram has a radially averaged power spectrum (RAPS) as shown in Figure 1.

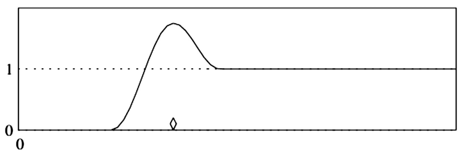


Figure 1: *Ideal blue noise radially averaged power spectrum.*

Widely studied in the planar domain since the well-known Dart Throwing [Coo86], blue noise sampling and Poisson disk distribution are now further investigated for surface meshes.

Related works

The blue noise sampling of surfaces can be done according to three ways. The first one consists in parameterizing the surface to a planar domain, and then in applying a 2D sampling pattern on it [AMD02, LLLF08, LWSF10]. It makes the sampling process easier but the output sampling may suffer from distortions when mapping back to 3D domain. The second approach is to make a Poisson disk sampling *directly* on the surface meshes [FZ08, CJW*09, BWWM10, ZGW11, CCS12]. These methods are all inspired by 2D sampling algorithms based on dart throwing [DH06,LD07,WCE07,Wei08,EDP*11]. They generate sampling patterns with typical blue noise properties, but they are sometimes time-consuming, especially when using geodesic distances. The third category is based on relaxation algorithms. The recent [XHGL12] is an extension of 2D sampling techniques [BSD09,XLGG11], and improves the well-known Lloyd's algorithm [Llo82] that greatly influences the blue noise properties of the results. Those methods overcome the problem of controlling the number of samples, unlike the dart throwing algorithms.

DOI: 10.2312/conf/EG2013/short/009-912

Motivation and contributions

Our aim is to develop an efficient and fast direct blue noise sampling algorithm for surface meshes. Especially, to efficiently sample CAD models and to produce meshes having good blue noise properties. We naturally avoid the use of 2D parameterizations, not suitable to deal with such meshes.

Among the direct sampling methods for surfaces, the method of [FZ08] computes exact isolines, which produces accurate but time-consuming sampling. It computes sphere boundaries around each new thrown sample, and then updates the connectivity of the available part of the mesh. [CJW*09] proposes a trade-off between accuracy and computation time by approximating both geodesic distances and available boundaries, by subdividing triangles that are partially contained into a sphere and by removing totally covered sub-triangles from available part of the mesh for the dart throwing, which may not be very efficient. In parallel, two Poisson disk sampling techniques take into account geometry constraints such as sharp features, either by using tiles [CCS12] or by directly sampling on the surface [ZGW11]. But both use Euclidean distances at the collision detection step, which is not convenient to achieve our objectives.

Inspired by [FZ08] and [ZGW11], we propose an efficient dart throwing for surface meshes that includes:

- a segmentation step to detect sharp features, in order to preserve them during sampling;
- an original formulation for the sphere radius computation, based on minimum and maximum curvature values;
- a fast and accurate geodesic estimation in order to speed-up the collision detection step.

2. Overview of our method

Let M be an input 2-manifold triangular mesh, of any genus, closed or not, and $M_{sampled}$ be the output sampled mesh. Our algorithm can be summarized in four steps, as follows:

1. Midpoint subdivision of the input mesh M yielding a densely sampled mesh M_{sub};
2. Classification of the vertices of M_{sub} with respect to three categories: *corners*, *sharp features* or *smooth regions* (Section 4);
3. Improved dart throwing on M_{sub} (Section 5);
4. (Optional) Triangulation of the samples to produce the mesh $M_{sampled}$.

3. Geodesic computation

Contrary to [FZ08] that uses time-consuming exact isolines on the input surface to compute geodesics, we rather resort to the well-known approximating algorithm of Dijkstra [Dij59], whose main benefit is its low-complexity. To enhance its accuracy, a midpoint subdivision of the input mesh M is first performed. This subdivided version M_{sub} will be the input for the rest of our algorithm. By this way, our algorithm tends to compute more precise geodesics.

4. Classification

Once the mesh M_{sub} is obtained, its vertices are separately classified into three categories: *corners*, *sharp features*, or *smooth regions*. Afterwards, the sampling will be driven by this classification, by successively generating samples on *corners*, on the *sharp features*, and finally on the *smooth regions*.

The classification is based on the normal tensor voting theory described in [KCL09]. It computes for each vertex v of M_{sub} a 3×3 weighted covariance matrix. Its three sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are used to classify v (λ_2 and λ_3 are the maximum and the minimum curvature values at v, respectively):

- If λ_1 is dominant, and λ_2, λ_3 are close to 0, v is classified into *smooth regions*;
- If λ_1 and λ_2 are dominant, and λ_3 is close to 0, v is classified into *sharp features*;
- If the three eigenvalues are approximately equal, v is classified into *corners*.

Our method is as efficient as the work of [FZ08,ZGW11], but it is faster because our classification does not require to link the feature vertices so as to generate feature lines [YLL*09].

5. Improved dart throwing

Any direct dart throwing technique for surfaces consists in selecting a point p on the mesh randomly, and considering it as a valid sample only if it fulfills the minimum distance requirement with respect to the samples already accepted. The minimum distance is relative to the radius R of a sphere associated to each sample. The main characteristics of our dart throwing algorithm are described below.

First, our dart throwing algorithm limits its sampling to the set of vertices of M_{sub}. This solution reduces the complexity of our algorithm, and we will prove that this limitation does not influence the sampling quality with respect to the blue noise properties.

Second, as introduced in Section 4, our algorithm successively distributes samples on *corners*, on *sharp features*, then on *smooth regions* to preserve geometrical features of M.

Finally, our dart throwing is done adaptively, according to the curvature. The radius R associated to a sample s depends on the two curvature values computed at its location on M_{sub}:

$$R = R_{min} \times (e^{C_{max} \times \lambda_2} + e^{C_{min} \times \lambda_3}),$$

with

$$R_{min} = \alpha \times \sqrt{\frac{|M|}{\Pi \times N_{target}}}.$$

N_{target} is the user-given target number of samples. $|M|$ corresponds to the area of M. C_{min} and C_{max} are negative parameters that weight the curvatures along each principal direction, whereas α can be used to shift the domain of the radius.

Empirically, we observed that $\alpha = 0.65$, $C_{min} = C_{max} = 0.0$ for the category *corners*, $C_{min} = C_{max} = -8.0$ for *sharp features*, and $C_{min} = C_{max} = -6.0$ for *smooth regions* ensure good blue noise properties, for all the tested models. Actually, only one input parameter is required, unlike [ZGW11]: the expected number of samples N_{target}, which makes our algorithm more convenient for users.

Globally, our proposal is the following. We first initialize a list of vertices available for the sampling (the so-called *candidate vertices*) with all the vertices of M_{sub}. After each valid dart throwing giving a valid sample s, the candidate vertices located into the sphere relative to s (depending on its radius R) are removed from the set of *available vertices*. If a dart throwing fails for a sample s, this latter is removed from *available vertices* to avoid a useless second test at this location. To efficiently evaluate the region of the mesh relative to a given sphere associated to a sample s, our Dijkstra-based algorithm is able to stop its region growing process if:

- one encountered vertex is not in the set of *available vertices*, revealing a "sphere collision". In that case, s is rejected;
- the geodesic value of the radius associated to s is reached. In that case, s is valid, so all the encountered vertices are removed from the set of *available vertices*.

6. Experimental Results and Discussion

In this section we evaluate the efficiency of our algorithm by sampling several CAD models of different genus. First, we verify that our sampling method preserves geometrical features. Figure 2 depicts three meshes sampled with our technique, and then triangulated. We clearly see that our algorithm preserves the sharp features automatically, and also deals with high genus models rapidly. For instance, the classification and our dart throwing of the input SOCKET mesh (right model in Figure 2) ran in 153 milliseconds on a PC (Intel Core i3 CPU 2.30 GHz, 4 GB RAM). Second, we use the differential analysis tool proposed by [WW11], dedicated to non-uniformly sampled surfaces, to analyse the blue noise properties produced by our sampling method. Figure 3 depicts from left to right, respectively the 2D periodogram, the RAPS and the anisotropy for several models sampled with our technique. For each model, the RAPS and the anisotropy are averaged over a set of 8 meshes, which have a number of vertices ranging from 1022 to 1062. Also, to evaluate the influence of the preservation of sharp features over the sampling properties, we draw the RAPS and the anisotropy obtained with (red curves) or without (blue curves) the classification step. Globally, our sampling technique preserves features while keeping satisfactory blue noise properties. In term of RAPS, we see that the curves are very similar with or without feature preservation, and have the typical variations of an "ideal" blue noise sampling (see Figure 1). Nevertheless, we observe that the genus of the input meshes influences the sharp transition of the RAPS.

The analysis of the sampling anisotropy is also satisfactory, since the curves tend to be horizontal beyond the cut-off frequency (the anisotropy values inferior to the cut-off frequency must not be considered as explained in [WW11]). Moreover, we have verified that the anisotropy always tends to the "ideal" anisotropy value according to the formula given in [WW11] (not shown on these figures), whatever the sampling technique used. We also observe that the anisotropy is higher than for smooth models (see [XHGL12] for comparison). It was expected since geometrical constraints along features must impose some specific directions on the sampled surfaces, which unavoidably reduces the isotropy of the sampled mesh. As an example, we can observe that the anisotropy of SOCKET does not tend to the same value with or without the feature preserving technique, because the high number of sharp features. These regions require more samples to the detriment of other regions, which globally increases the sampling anisotropy. Anyway, the anisotropy shows satisfactory blue noise properties even with this model.

7. Conclusion

We presented a fast, automatic and direct blue noise sampling technique for surface meshes, that is able to handle models with sharp features, but also models of high genus. We showed that it is possible to preserve sharp features of surface meshes during sampling while satisfying good blue noise properties. One of our future works is to analyze in depth the quality and the fidelity of the meshes obtained with the remesher based on our sampling scheme (triangle ratio, remeshing error, visual quality...).

8. Acknowledgements

This work is supported by a grant from *Région Provence Alpes Côte d'Azur* and Noomeo Corporation (France).

References

[AMD02] ALLIEZ P., MEYER M., DESBRUN M.: Interactive geometry remeshing. In *Proceedings of ACM SIGGRAPH'02* (2002), pp. 347–354. 1

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of lloyd's method. *ACM Transactions on Graphics 28*, 3 (2009), 86:1–8. 1

[BWWM10] BOWERS J., WANG R., WEI L.-Y., MALETZ D.: Parallel poisson disk sampling with spectrum analysis on surfaces. *ACM Trans. Graph. 29*, 6 (Dec. 2010), 166:1–166:10. 1

[CCS12] CORSINI M., CIGNONI P., SCOPIGNO R.: Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transaction on Visualization and Computer Graphics 18*, 6 (2012), 914–924. 1, 2

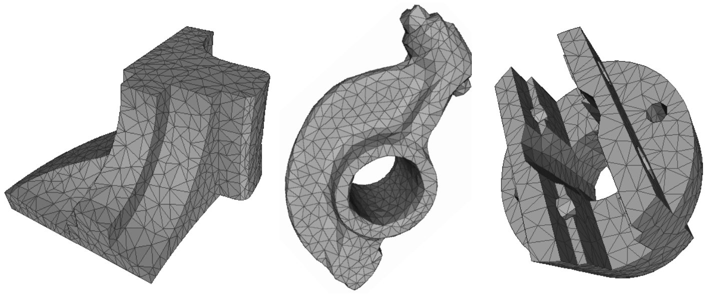
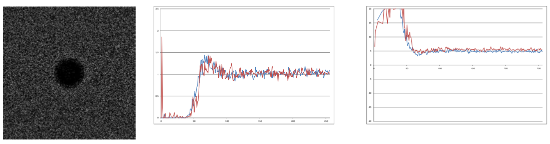
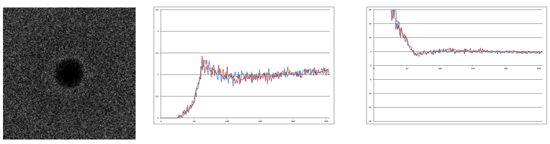


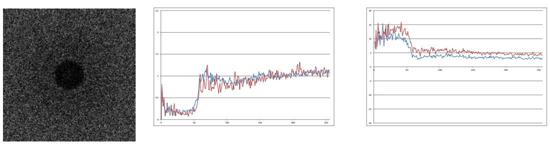
Figure 2: *Three CAD models sampled with our method, and then triangulated. From left to right:* FANDISK *(genus 0; input mesh : 6,475 vertices; output mesh: 1,062 vertices),* ROCKER ARM *(genus 1; input mesh: 10,000 vertices; output mesh: 1,055 vertices), and* SOCKET *(genus 7; input mesh: 836 vertices; output mesh: 1,024 vertices).*



(a) FANDISK.



(b) ROCKER ARM.



(c) SOCKET.

Figure 3: *Analysis of the sampling quality for three CAD models, with (red curves) or without (blue curves) the proposed classification step. From left to right: periodogram, RAPS, anisotropy.*

[CJW*09] CLINE D., JESCHKE S., WHITE K., RAZDAN A., WONKA P.: Dart throwing on surfaces. *Computer Graphics Forum 28*, 4 (6 2009), 1217–1226. 1, 2

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Transactions on Graphics 5*, 1 (Jan. 1986), 51–72. 1

[Cro77] CROW F. C.: The aliasing problem in computer-generated shaded images. *Commun. ACM 20*, 11 (Nov. 1977), 799–805. 1

[DH06] DUNBAR D., HUMPHREYS G.: A spatial data structure for fast poisson-disk sample generation. In *ACM SIGGRAPH 2006 Papers* (2006), SIGGRAPH '06, pp. 503–508. 1

[Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271. 2

[EDP*11] EBEIDA M. S., DAVIDSON A. A., PATNEY A., KNUPP P. M., MITCHELL S. A., OWENS J. D.: Efficient maximal poisson-disk sampling. *ACM Transactions on Graphics 30*, 4 (July 2011), 49:1–49:12. 1

[FZ08] FU Y., ZHOU B.: Direct sampling on surfaces for high quality remeshing. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling* (New York, NY, USA, 2008), SPM '08, ACM, pp. 115–124. 1, 2

[KCL09] KIM H. S., CHOI H. K., LEE K. H.: Feature detection of triangular meshes based on tensor voting theory. *Computer Aided Design 41*, 1 (Jan. 2009), 47–58. 2

[LD07] LAGAE A., DUTRE P.: A comparison of methods for generating poisson disk distributions. *Computer Graphics forum* (2007). 1

[LLLF08] LI H., LO K.-Y., LEUNG M.-K., FU C.-W.: Dual poisson-disk tiling: An efficient method for distributing features on arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics 14*, 5 (2008), 982–998. 1

[Llo82] LLOYD S.: Least squares quantization in pcm. *IEEE Trans. Inf. Theor. 28*, 2 (september 1982), 129–137. 1

[LWSF10] LI H., WEI L.-Y., SANDER P. V., FU C.-W.: Anisotropic blue noise sampling. In *ACM SIGGRAPH Asia 2010 papers* (2010), SIGGRAPH ASIA '10, ACM, pp. 167:1–167:12. 1

[WCE07] WHITE K. B., CLINE D., EGBERT P. K.: Poisson disk point sets by hierarchical dart throwing. *Symposium on Interactive Ray Tracing 0* (2007), 129–132. 1

[Wei08] WEI L.-Y.: Parallel poisson disk sampling. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 20:1–20:9. 1

[WW11] WEI L.-Y., WANG R.: Differential domain analysis for non-uniform sampling. *ACM Trans. Graph. 30*, 4 (July 2011), 50:1–50:10. 3

[XHGL12] XU Y., HU R., GOTSMAN C., LIU L.: Blue noise sampling of surfaces. *Computer and Graphics* (2012). 1, 3

[XLGG11] XU Y., LIU L., GOTSMAN C., GORTLER S. J.: Capacity-constrained delaunay triangulation for point distributions. *Computers & Graphics 35*, 3 (2011), 510–516. 1

[YLL*09] YAN D.-M., LÉVY B., LIU Y., SUN F., WANG W.: Isotropic remeshing with fast and exact computation of restricted voronoi diagram. In *Proceedings of the Symposium on Geometry Processing* (2009), SGP '09, pp. 1445–1454. 2

[ZGW11] ZHANG H., GENG B., WANG G.: Approximate poisson disk sampling on mesh. *Journal of Computer-Aided Design and Computer Graphics 23*, 1 (2011), 62–69. 1, 2, 3