



**HAL**  
open science

## Cache-friendly micro-jittered sampling

Arthur Dufay, Pascal Lecocq, Romain Pacanowski, Jean-Eudes Marvie,  
Xavier Granier

► **To cite this version:**

Arthur Dufay, Pascal Lecocq, Romain Pacanowski, Jean-Eudes Marvie, Xavier Granier. Cache-friendly micro-jittered sampling. SIGGRAPH 2016, Jul 2016, Anaheim, United States. 10.1145/2897839.2927392 . hal-01325702

**HAL Id: hal-01325702**

**<https://inria.hal.science/hal-01325702>**

Submitted on 2 Jun 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Cache-friendly micro-jittered sampling

Arthur Dufay\* †‡

Pascal Lecocq\* †

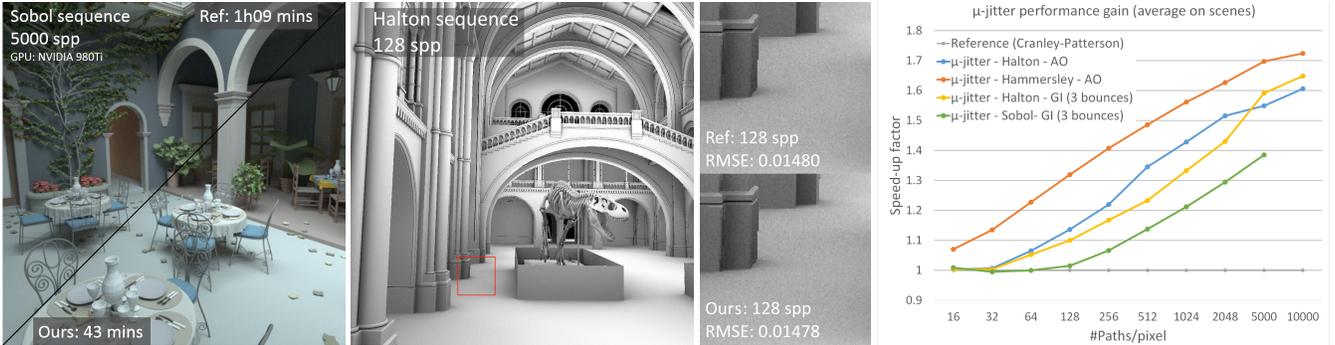
Romain Pacanowski ‡

Jean-Eudes Marvie †

Xavier Granier ‡

†Technicolor Research & Innovation

‡LP2N (UMR 5298: IOGS - Univ. Bordeaux - CNRS) - Inria, F33400 Talence



**Figure 1:** Rendering times comparison (left) on the path-tracing-algorithm between a Cranley-Patterson de-correlation method as reference and our micro-jittering approach on the San-Miguel scene (10,495,880 tris, 3 bounces/path). Our method reduces substantially the rendering times for equal numbers of paths per pixel (right), while preserving the visual convergence of any low-discrepancy sequence (middle).

## 1 Motivation

Monte-Carlo integration techniques for global illumination are popular on GPUs thanks to their massive parallel architecture, but efficient implementation remains challenging. The use of randomly de-correlated low-discrepancy sequences in the path-tracing algorithm allows faster visual convergence. However, the parallel tracing of incoherent rays often results in poor memory cache utilization, reducing the ray bandwidth efficiency. Interleaved sampling [Keller et al. 2001] partially solves this problem, by using a small set of distributions split in coherent ray-tracing passes, but the solution is prone to structured noise. On the other hand, ray-reordering methods [Pharr et al. 1997] group stochastic rays into coherent ray packets but their implementation add an additional sorting cost on the GPU [Moon et al. 2010] [Garanzha and Loop 2010].

We introduce a micro-jittering technique for faster multi-dimensional Monte-Carlo integration in ray-based rendering engines. Our method, improves ray coherency between GPU threads using a slightly altered low-discrepancy sequence rather than using ray-reordering methods. Compatible with any low-discrepancy sequence and independent of the importance sampling strategy, our method achieves comparable visual quality with classic de-correlation methods, like Cranley-Patterson rotation [Kollig and Keller 2002], while reducing rendering times in all scenarios.

**Keywords:** path-tracing, ray coherency, sampling, jittering

**Concepts:** •Computing methodologies → Ray tracing;

\* Joint first authors.

† firstname.lastname@technicolor.com

‡ firstname.lastname@institutoptique.fr

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). © 2016 Copyright held by the owner/author(s). SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,

ISBN: 978-1-4503-4282-7/16/07

DOI: <http://dx.doi.org/10.1145/2897839.2927392>

## 2 Micro-jittered sampling

Our approach leverages the coherency naturally present in 3D scenes with constrained Cranley-Patterson rotation. Cranley-Patterson rotation de-correlates samples by adding a random offset in each dimension of a single random sequence. However, the parallel evaluation of the  $i$ -th ray sample within a block of threads often results in random access to 3D space region, causing memory cache misses.

**Core idea.** Because close pixels have higher probability to share the same visible surface characteristics, the parallel tracing of nearly similar rays on first bounces tends to follow similar paths in space, increasing thread data coherency. Based on this simple observation, our micro-jittering strategy consists in restricting Cranley-Patterson rotation to micro-jitters within a small volume of size  $\mu$  using a signed random uniform distribution  $\xi$ . (eq.1).

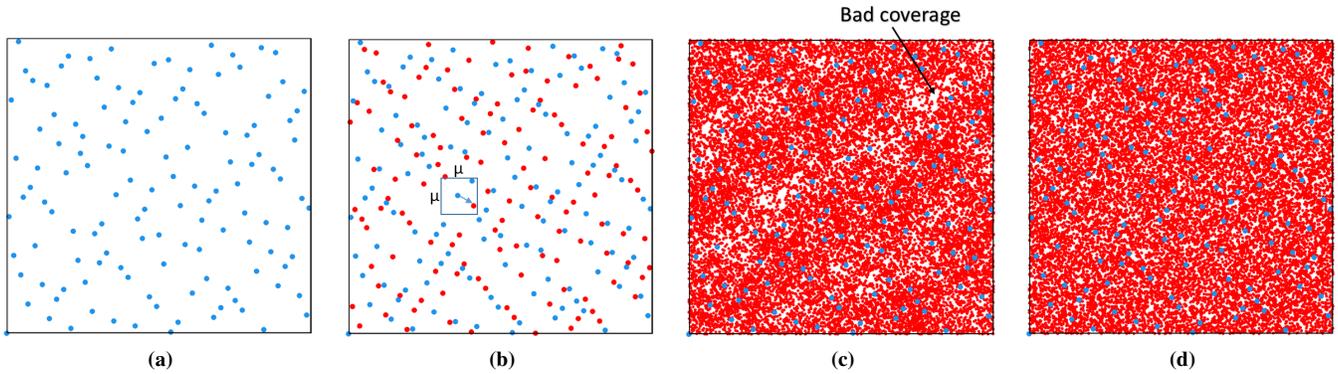
$$y_i = x_i + \mu_{(seq, N)} \xi, \quad \xi \in [-0.5, 0.5]^s \quad (1)$$

To prevent the apparition of structured noise, or any bias,  $\mu$  is adaptively chosen to ensure that the de-correlated sequences uniformly draw samples within the unit hypercube see Figure 2. This depends on a constant  $K$  representative of the star-discrepancy of the random sequence, and  $N$ , the sampling count.

$$\mu_{(seq, N)} = K_{D_{seq}^*} N^{-1/s} \quad (2)$$

As the result, each indexed sample in each de-correlated sequence orbits around its original position, increasing the probability to address the same region of space while preserving (mostly) the discrepancy property of the sequence.

Our method shares some similarities with the *uniform jitter sampling* method described by [Ramamoorthi et al. 2012] and devised for area-light visibility computations. However our approach is somewhat different. We focus on performance rather than visibility and preserve the noise property by micro-jitter random sequences in a domain based on their discrepancy characteristics.



**Figure 2:** Illustration of the micro-jittered approach: 128 samples (blue dots) of a two-dimensional Halton sequence (a) de-correlated by micro-jittering  $1\times$  (red dots) and  $150\times$  (c,d) within a small area of size  $\mu$ . Setting  $\mu$  too small may result in a bad coverage of the de-correlated integration domain (c). To prevent any bias or structured noise, we adaptively choose  $\mu$  according to the sampling count and the star-discrepancy of the sequence to evenly cover the domain.

### 3 Results

We implemented the micro-jittering approach in an in-house GPU path-tracer and in a modified version of Blender Cycles (GPU and CPU). We compared our approach against Cranley-Patterson rotation method in terms of visual quality and rendering performances. Comparisons were done on 3D scenes of various complexity, several low-discrepancy sequences and different sampling count. In all scenarios, our method delivers superior rendering performances while providing comparable visual quality. As the sampling rate increases, our method perform faster, converging to coherent path-tracing performances without its inherent flaw. In some scenarios, our method can achieve speed-up factors up to  $2\times$ . We also successfully applied our method to screen-space rendering techniques. In a ray-marched SSAO implementation, our micro-jittering approach also improves rendering times without altering the visual quality.

Our method is simple, easy to implement and can drastically reduce rendering times for high-demanding stochastic ray-tracing applications with random memory accesses.

### 4 Acknowledgments

We would like to thank Gaël Sourimant for its implementation and experimentations of the micro-jittering method in a SSAO context. We also thank the anonymous reviewers for their valuable comments and feedback.

Scene credits: the San-Miguel scene in teaser-left was modeled by Guillermo M. Leal Llaguno. The London NH museum in middle was modeled by Alvaro Luna Bautista and Joel Andersdon. The interior 3D model from figure ?? is coming from the chocofur store ([store.chocofur.com](http://store.chocofur.com)).

### References

GARANZHA, K., AND LOOP, C. 2010. Fast ray sorting and breadth-first packet traversal for gpu ray tracing. In *Computer Graphics Forum*, vol. 29, Wiley Online Library, 289298.

KELLER, A., KELLER, E., AND HEIDRICH, W. 2001. Interleaved sampling. In *Rendering Techniques 2001 (Proc. 12th Eurographics Workshop on Rendering)*, Springer, 269–276.

KOLLIG, T., AND KELLER, A. 2002. Efficient multidimensional sampling. In *Computer Graphics Forum*, vol. 21, Wiley Online Library, 557–563.

MOON, B., BYUN, Y., KIM, T.-J., CLAUDIO, P., KIM, H.-S., BAN, Y.-J., NAM, S. W., AND YOON, S.-E. 2010. Cache-oblivious ray reordering. *ACM Trans. Graph.* 29, 3 (July), 28:1–28:10.

PHARR, M., KOLB, C., GERSHBEIN, R., AND HANRAHAN, P. 1997. Rendering complex scenes with memory-coherent ray tracing. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '97, 101–108.

RAMAMOORTHY, R., ANDERSON, J., MEYER, M., AND NOWROUZEZAHRAI, D. 2012. A theory of monte carlo visibility sampling. *ACM Trans. Graph.* 31, 5 (Sept.), 121:1–121:16.



**Figure 3:** Interior scene rendered with 10000 paths/pixel, 4 diffuse/glossy bounces and 24 transmissions bounces using a sobol sequence. The reference image was generated using a classic Cranley-Patterson de-correlation method. The image referenced as "ours" was generated using our micro-jittered sampling technique. Both rendering times are reported from a NVIDIA 980Ti GPU.