Evaluation of sampling techniques with BM3D denoising for Monte Carlo rendering

Kshitij Adlakha
Stanford University
kadlakha@stanford.edu

Abstract

BM3D is a denoising method based on the fact that an image has a locally sparse representation in transfer domain. This sparsity is enhanced by grouping similar 2D image patches into 3D groups. In this paper we look many common sampling techniques used in modern ray tracing software, and evaluate their compatibility with BM3D denoising technique.

1. Introduction

Synthesizing realistic imaging from virtual scene models requires simulating the traversal of light from light sources through the scene, onto an image sensor. Modelling the physics involved in light transport through complex geometry scenes is not feasible to solve empirically to render the scene. As a result all modern ray tracers use Monte Carlo rendering techniques. This provides an elegant solution to randomly sample the integrand of the rendering equation by tracing random light paths from the camera towards the light sources. From each path a color value is calculated, called a sample.

Even though an image could be generated by just sampling the function precisely at the pixel positions, a better result can be obtained by taking more samples at different positions and incorporating this additional information about the image function into the final pixel values. As sampling involves approximation, it can introduce aliasing which can manifest itself as jagged edges. Various sampling techniques have been developed to alleviate this shortcoming of sampling, by modifying the random sampling in a way that post rendering denoising can reconstruct an image close to the ground truth.

Over the years, many denoising techniques have been developed to effectively remove the 'noise' that occurs from insufficient sampling. One landmark denoising algorithm that was developed in 2007 was BM3D (Block-matching and 3D filtering), which employs non-local means filtering by grouping 2D blocks on the image into a 3D array. These groups are collaboratively filtered by moving the group to transfer-domain, where the signal is sparse ans noise can be removed, followed by inverse 3D transformation.

In this paper we evaluate compatibility of many different commonly used sampling techniques in modern ray tracing software, with BM3D denoising.

2. Related Work

Over the decades a lot of effort has been put into finding effective sampling techniques for ray tracers. Crow [3] first identified aliasing as a major source of artifacts in computer-generated images. Using nonuniform sampling to turn aliasing into noise was introduced by Cook [2] and Dippé and Wold [5]. Lee, Redner, and Uselton [12] developed a technique for adaptive sampling based on statistical tests that computed images to a given error tolerance. Mitchell investigated sampling patterns for ray tracing extensively. His 1987 and 1991 SIGGRAPH papers on this topic have many key insights. Mitchell [15] investigated how much better stratified sampling patterns are than random patterns in practice. The (0,2)-sequence sampling techniques are based on a paper by Kollig and Keller [10]. The Maximized Minimal Distance Sampler is based on generator matrices found by Grünschloß and collaborators [6] [7]. Sobol [16] introduced the family of generator matrices used in Sobol's Sampler.

The implementation of BM3D have been ported over to the GPU using CUDA [8] [4], providing significant parallel performance. This makes BM3D an suitable denoising candidate to provide real time ray traced content with low sample per pixel count.

In recent years machine learning based techniques have overtaken the traditional denoiser like BM3D. These machine learning or deep learning techniques generally require a ground truth image to train the network, which is not realistic to acquire especially for content that requires real time rendering. Methods have been developed though that can denoise with requiring ground truth images [13].

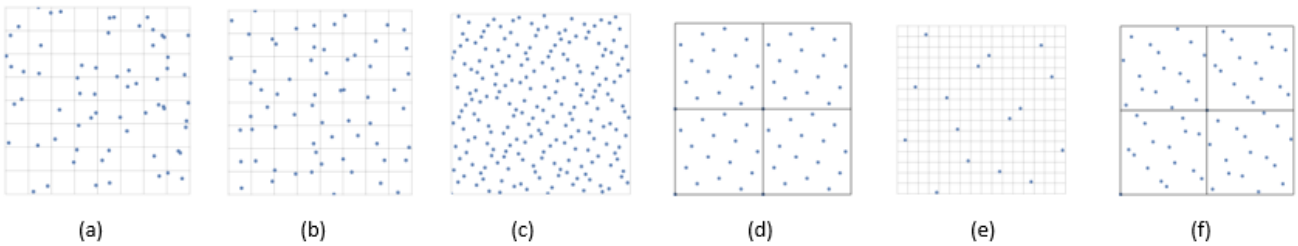

Figure 1. Example distribution of samples in the sampling techniques under evaluation (a) random (b) stratified (c) halton (d) MaxMin Distance (e) (0,2) sequence (f) Sobol

3. BM3D denoising

BM3D is a image denoising technique based on an enhanced sparse representation in transform domain. The transform-domain denoising methods typically assume that the true signal can be well approximated by a linear combination of few basis elements. That is, the signal is sparsely represented in the transform domain. So by preserving the high magnitude transform co-efficients, and discarding others which are likely due to noise, we can come close to recovering the original signal. The effectiveness of this depends on the sparsity we can achieve from the transform, and the properties of the original signal. If we can get high similarity between the grouped blocks, the transform can achieve a highly sparse representation of the true signal so that the noise can be well separated by shrinkage. In this way, the collaborative filtering reveals even the finest details shared by grouped fragments and at the same time it preserves the essential unique features of each individual fragment.

4. Sampling

In order to compute the discrete pixel values in the digital image, it is necessary to sample the original continuously defined image function. In ray tracing systems this done by tracing individual rays at sample positions. While an image could be generated by just sampling the function precisely at the pixel positions, a better result can be obtained by taking more samples at different positions and incorporating this additional information about the image function into the final pixel values. Because the sampling and reconstruction process involves approximation, it introduces error known as aliasing, which can manifest itself in many ways, including jagged edges or flickering in animations. These errors can occur because the sampling process cannot capture all of the information from the continuously defined image function.

The most basic form of sampling is random sampling, where samples are randomly picked in a pixel region. The methods is rarely used because it is more prudent to expend samples in areas of the scenes with higher amount of detail. Many additional sampling techniques have been developed over the decades that can achieve that goal. A couple of them are discussed below.

4.1. Stratified Sampler

In stratified sampling we subdivide pixel areas into rectangular regions and generates a single sample inside each region. Each region is called a strata. The key idea behind stratification is that by subdividing the sampling domain into non-overlapping regions and taking a single sample from each one, we are less likely to miss important features of the image entirely, since the samples are guaranteed not to all be close together. Within each strata, the sample is places randomly by jittering the center point of the stratum. The non-uniformity that results from this jittering helps turn aliasing into noise.

4.2. Halton Sampler

Halton sampler is based on algorithms that directly generate low-discrepancy point sets. Unlike the points generated by the Stratifed sampler, the Halton sampler not only generates points that are guaranteed to not clump too closely together, but it also generates points that are simultaneously well distributed over all of the dimensions of the sample vector—not just one or two dimensions at a time, as the Stratified sampler did. One of the most useful characteristics of the Halton sequence is that it can be used even if the total number of samples needed isn't known in advance; all prefixes of the sequence are well distributed, so as additional samples are added to the sequence low discrepancy will be maintained.

4.3. (0, 2)-Sequence Sampler

This sampling technique takes advantage of a remarkable property of certain low-discrepancy sequences that allows us to satisfy two desirable properties of samples (only one of which was satisfied with the Stratified sampler): they

Sampler	random	halton	02sequence	maxmindist	sobol	stratified
1 spp	15.00	14.70	15.00	17.99	14.78	15.01
2 spp	17.93	17.61	18.25	20.17	17.36	18.07
4 spp	20.61	20.27	20.97	21.62	19.32	20.79

Figure 2. Geometric mean of PSNR values for the scenes using the 6 sampling techniques.

generate sample vectors for a pixel's worth of image samples such that the sample values for each pixel sample are well distributed with respect to each other, and simultaneously such that the aggregate collection of sample values for all of the pixel samples in the pixel are collectively well distributed. This sequence uses the first two dimensions of a low-discrepancy sequence derived by Sobol . This sequence is a special type of low-discrepancy sequence known as a (0,2)-sequence. (0,2)-sequences are stratified in a very general way.

4.4. Maximized Minimal Distance Sampler

The (0,2)-sequence sampler is more effective than the stratified sampler, thanks to being stratified over all elementary intervals. However, it still sometimes generates sample points that are close together. An alternative is to use a different pair of generator matrices that not only generate (0,2)-sequences but that are also specially designed to maximize the distance between samples; this approach is implemented by the Maximized Minimal Distance Sampler. There are 17 of these specialized matrices, one for each power-of-two number of samples.

4.5. Sobol Sampler

Sobol Sampler is based on a series of generator matrices due to Sobol. The samples from the sequence that these matrices generate are distinguished by both being very efficient to implement—thanks to being entirely based on base-2 computations—while also being extremely well distributed over all dimensions of the sample vector. he weakness of the Sobol points is that they are prone to structural grid artifacts before convergence.

5. Method

To analyze the effectiveness of sampling methods with BM3D, we generate ray traced images with low sample count and run them through the denoiser. To generate ray traced images we use PBRT [14] software package. There a large set of scenes available for PBRT [17] [1]. 10 scenes were picked to cover the variety of scenes available for the simulator. PBRT was run on these scenes with the 6 different samplers described in Section 2.

Each of the samplers was run with 3 samples per pixel settings : 1,2 and 4. These 3 values should cover the real-

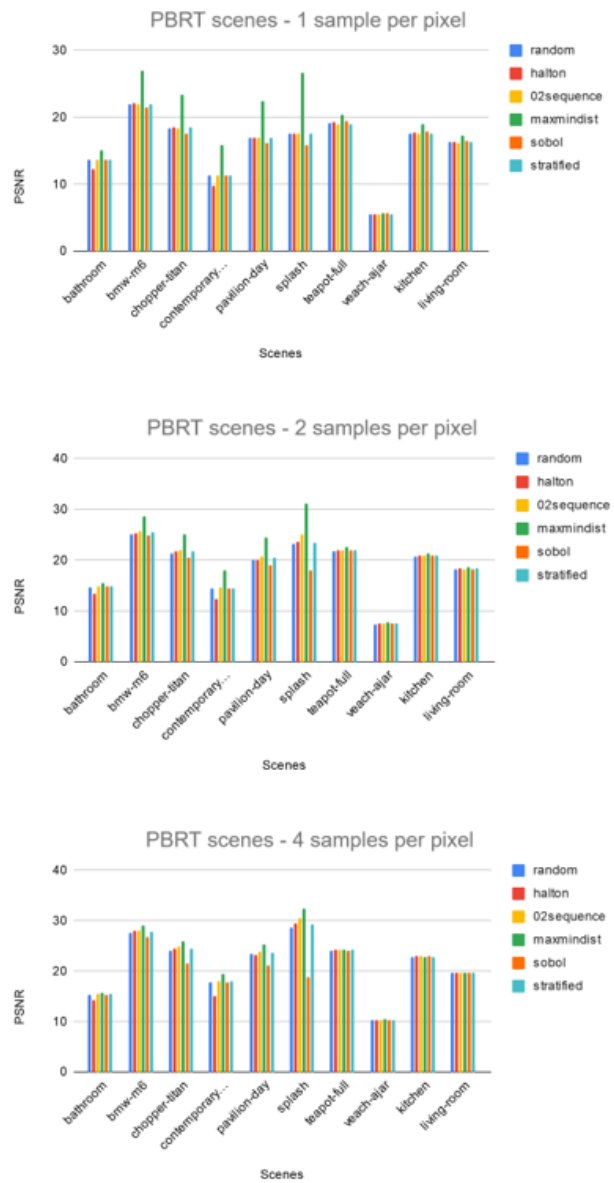

Figure 3. Plot of the PSNR values for different scenes using the 6 sampling techniques.

istic number of rays that can be traced in a ray tracing system with hardware acceleration for real time ray-tracing. To generate the ground truth image the scene was run on PBRT with a very high sample count.

The images generated from PBRT we fed into the BM3D denoiser. The MATLAB implementation of BM3D denoiser [11] was used, which has been highly optimized. The denoised image for each configuration was gathered and compared against the ground truth image to calculate the

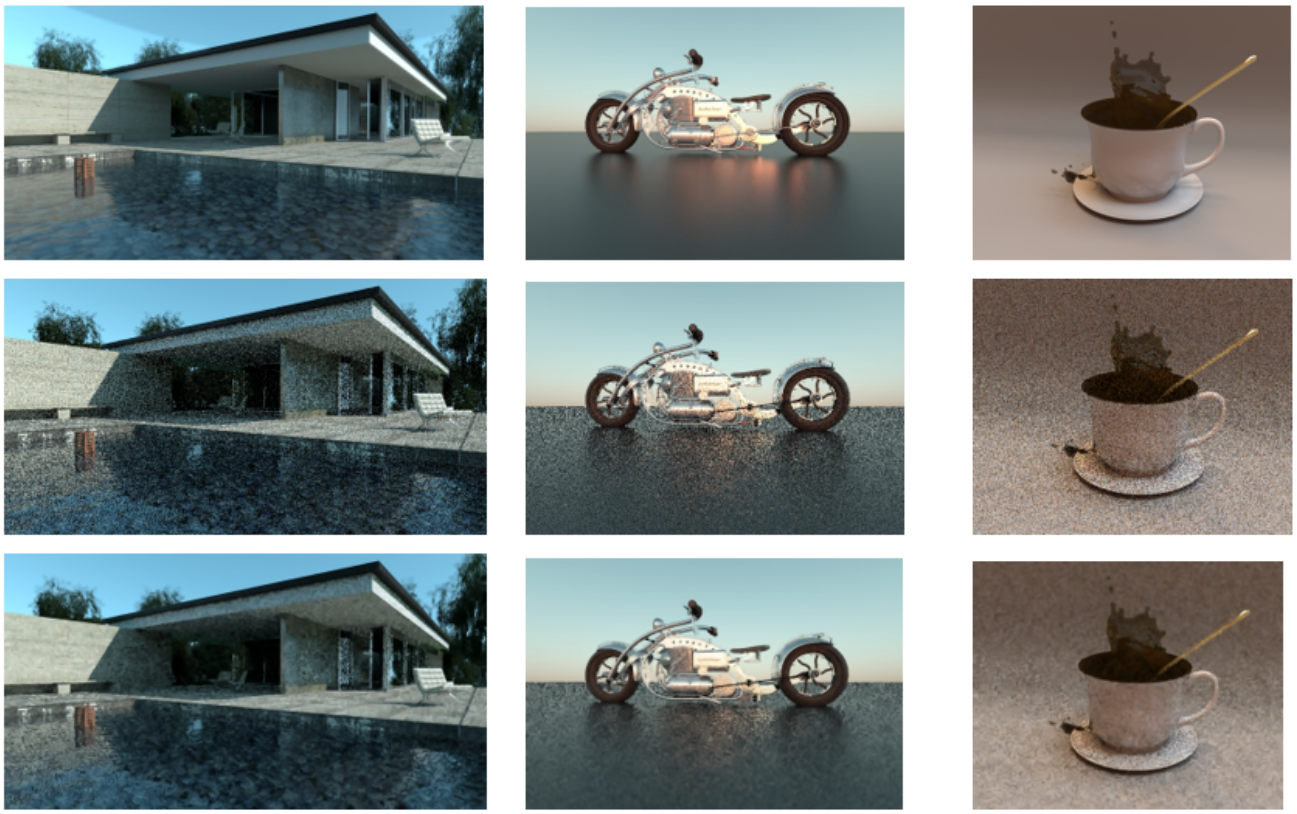

Figure 4. Results from denoising 3 scenes (from left, pavilion-day, chopper-titan and splash) with MaxMinDist sampler and using 1 sample per pixel

PSNR. The PSNR value was used to compare the quality of denoise for each sampler.

6. Results

When running denoising on images generated with 1 sample per pixel, MaxMinDist sampler is significantly ahead in PSNR for 5 out of the 10 scenes. For 2 other scenes MaxMinDist is slightly ahead of the competition.

When using 2 samples per pixel the gains of the MaxMinDist sampler go down, but is still 10% above other samplers. At 4 sample per pixels all the samplers performance is close to each other, apart from Sobol sampler which suffers in the splash scene.

In all of the cases none of the other samplers break out from the competition apart from the MaxMinDist sampler. BM3D excels at images when it can create large group of similar blocks so images with more regular geometry do better. The veach-ajar scene performs poorly with all the samplers because of its unique property that there are no light sources in the scene apart from an ajar door which is letting outside light into the room.

7. Discussion and Future Work

The best results for BM3D is shown by Maximized Minimal Distance Sampler by a wide margin, especially for low sample per pixel count. The sampler has the property of being deterministic in its generation, and efficient in its implementation for powers of 2 samples. BM3D transforms the groups into a domain where the signal is sparse so that the noise can be separated. The MaxMinDist Sampler provides well separated samples while also not a lot of randomness, making it easier to separate out coefficients related to noise in the transform domain.

The BM3D implementation used in this work uses are uniform standard deviation of noise. There are algorithms [9] available to calculate the standard deviation of Monte Carlo noise in localized regions which could be applied to BM3D to improve its denoising.

To achieve real time rendering the sampling and denoising needs to run within the time constraints of the frame rate. CUDA based implementations already exist for BM3D, as discussed earlier in the paper. A GPU compatible implementation would also need to be developed for

the MaxMinDist sampler so that it can run efficiently in a parallel environment.

References

[1] B. Bitterli. Rendering resources, 2016. https://benedikt-bitterli.me/resources/.

[2] R. L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, Jan. 1986.

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

[4] A. Davy and T. Ehret. Gpu acceleration of nl-means, bm3d and vbm3d. *Journal of Real-Time Image Processing*, 02 2020.

[5] M. A. Z. Dippé and E. H. Wold. Antialiasing through stochastic sampling. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, page 69–78, New York, NY, USA, 1985. Association for Computing Machinery.

[6] L. Grünschloß, J. Hanika, R. Schwede, and A. Keller. (t, m, s)-nets and maximized minimum distance. In A. Keller, S. Heinrich, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pages 397–412, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[7] L. Grünschloß, M. Raab, and A. Keller. Enumerating quasi-monte carlo point sequences in elementary intervals. In L. Plaskota and H. Woźniakowski, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2010*, pages 399–408, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[8] D. Honzátko and M. Kruliš. Accelerating block-matching and 3d filtering method for image denoising on gpus. *Journal of Real-Time Image Processing*, pages 1–15, 11 2017.

[9] N. K. Kalantari and P. Sen. Removing the noise in monte carlo rendering with general image denoising algorithms. *Computer Graphics Forum*, 32(2pt1):93–102, 2013.

[10] T. Kollig and A. Keller. Efficient multidimensional sampling. In *Computer Graphics Forum*, volume 21, pages 557–563. Wiley Online Library, 2002.

[11] V. K. Kostadin Dabov, Alessandro Foi and K. Egiazarian. Image and video denoising by sparse 3d transform-domain collaborative filtering : Software, 2007. http://www.cs.tut.fi/ foi/GCF-BM3D/index.html.

[12] M. E. Lee, R. A. Redner, and S. P. Uselton. Statistically optimized sampling for distributed ray tracing. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, page 61–68, New York, NY, USA, 1985. Association for Computing Machinery.

[13] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2noise: Learning image restoration without clean data, 2018.

[14] W. J. Matt Pharr and G. Humphreys. Physically based rendering: From theory to implementation. `https://github.com/mmp/pbrt-v3`, Oct 2019.

[15] D. P. Mitchell. Consequences of stratified sampling in graphics. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 277–280, 1996.

[16] I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

[17] Various. Physically based rendering scenes. `https://pbrt.org/scenes-v3.html`, Oct 2019.