Discontinuity analysis for noise estimation in stochastic computations of photo-realistic soft shadows

Moiseenko Yuri
Yuri.Moiseenko@intel.com,
Sevastianov Igor
Igor.Sevastianov@intel.com,
Soupikov Alexei
Alexei.Soupikov@intel.com

Abstract

We present a novel method for area/linear light source visibility and lighting computation. Our algorithm computes illumination at a given point with specified quality. The method allows for considerable reduction of number of visibility tests while minimizing overall computation time using computationally inexpensive quality criteria estimation. Proposed approach is suitable for thread parallelization and using SIMD instructions.

Keywords: Area light source, linear light source, soft shadows, ray tracing, visibility, parallelization.

1. INTRODUCTION

Soft shadows is a must have feature of any photo-realistic rendering system. Soft shadows are traditionally considered as computationally expensive and there is a large number of papers published describing various approaches to make their computations cheaper. For example, in [3] authors consider applying various integration approaches to linear lighting computations. [4] considers different sampling strategies for soft-shadows computations from different light sources. Vast majority of approaches for computing soft shadows are based on numerical precision estimation as a quality measure of generated image. In fact, acceptable image quality is achievable with far less number of samples than numerical precision requires. As stated in [1], the main problem for any integration approach in computer graphics is unknown number and structure of discontinuities of an integrand (e.g. visibility function), which thus might have infinite variation and, for example, the superiority of quasi-Monte Carlo integration has not been theoretically justified. We are using discontinuity analysis as a proxy for noise estimation and predicting of noise influence in rendered image.

2. PROBLEM STATEMENT

Our goal is an algorithm that would

1. perform efficient computation of correct photo-realistic soft shadows,
2. maintain easy parallelization,
3. enable efficient usage of tracing multiple rays simultaneously

We suggest using overall number of visibility tests (ray-scene intersection or occlusion tests) as the measure of efficiency and a basis of comparison with other approaches. The goals of photo-realism allow using such image quality characteristics as soft shadow shape, brightness, brightness gradients, etc. as a measure of correctness.

3. IDEA

Idea of our method is to save number of traced rays (visibility tests) using area classification. For the specific area the algorithm we suggest is trying to predict its occlusion state with respect to the area light source: fully visible or fully occluded or partially visible. Additionally, in case of penumbra, the algorithm predicts noise level. Obviously, the computational cost of such estimation for prediction should be much less than cost of shooting additional samples. In case if ray tracing is fast the cost of estimator becomes very important issue. We discuss this later in Performance Issues chapter.

The specific choice of sampling strategy is also very important factor when minimizing noise, so we are suggesting stratified sampling by sub-dividing an area light into areas having equal form-factor with respect to a point of interest.

4. DEFINITIONS

In this section we'd like to define and briefly explain several important terms and assumptions we will be using further in this paper.

Form-factor (between object and point) – describes the fraction of energy which leaves object and arrives at a point.

Visibility test (between two points in 3D space) – function, that returns 1 if points 'see' each other, i.e. aren't occluded by any object in the scene, and 0 else.

Brute force method to compute soft-shadows is classic approach when we placing some fixed number of quasi-random samples on the light source to compute visibility and illumination.

Estimation computations are all computations that spent on estimation of illumination of a given point excluding the cost of visibility tests (i.e. excluding ray tracing). Mainly these are computations of predictor saving visibility tests, generating samples, etc.

Packet of points/rays – set of points/rays processed simultaneously by the presented algorithm.

For the sake of simplicity we will consider the case of linear light source in details and make remarks on application of the algorithm for area light sources.

Note, that in this paper we describe approach to estimate visibility value. In order to evaluate radiance at specific point, visibility estimation should be convolved with irradiance at this point.

All performance measurements have been performed on dual-CPU Intel® Xeon™ 3.05Ghz machine with 1GB of PC2100 DDR.

5. PERFORMANCE ISSUES

The question one might to ask is why bothering with new adaptive method of soft shadows computation? Performance gain from using any prediction and estimation procedure is possible only if the computational cost of such procedure is significantly less than cost of visibility tests saved due to prediction. Therefore the slower visibility tests (due to the local/global scene complexity and other reasons) the higher is relative performance gain. On the other hand, the better is performance of visibility test the smaller speed-up we obtain from any method including ours.

Ray tracing has evolved to the point where tens of millions of ray-scene intersection tests on a single CPU machine is a feasible number. To accomplish such performance one needs using a variety of architectural features of modern platforms such as SIMD instructions or caches. So a computationally efficient and architecture friendly estimation procedure is required to balance load between visibility tests and estimation computations.

6. HIERARCHICAL VISIBILITY CALCULATION

Idea of the hierarchical visibility calculation is to select start subdivision of the linear light source and produce further subdivision if needed. First, linear light source is subdivided into line segments in such a way that each line segment occupies equal angle relatively to the specific point. For each line segment we perform quasi random visibility tests and then we consider three cases:

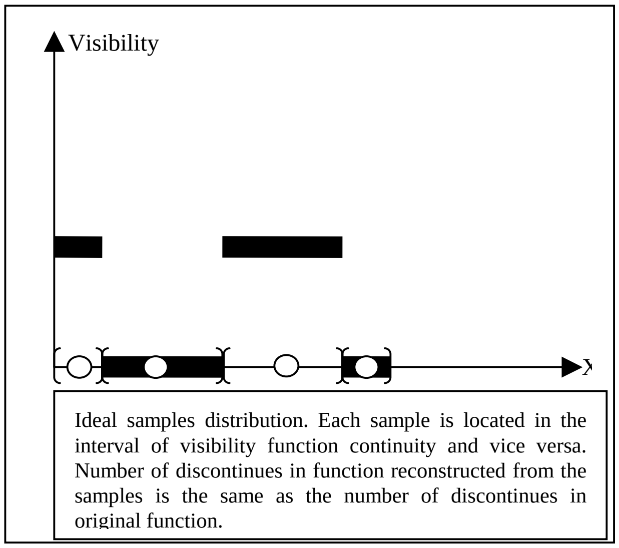
1. All visibility tests are non-occluded.
2. All visibility tests are occluded.
3. Some number of tests is occluded and other is not.

In the first case our method makes the decision that point is fully illuminated and in the second case - that point is not illuminated by the light source. In the third case we check if angle is not too small and then perform subdivision and recursively process new line segments. The subdivision maintains a property that each new line segment has equal angle relatively to specific point.

The algorithm performs calculations for a packet of points simultaneously using SIMD instructions. So the algorithm makes one single decision for all points in the packet. For this purpose we select pivot point – a single point in the packet that drives decision process. The computations related to prediction are performed for this point. Among all points in the packet we select one as a pivot point basing on the maximum form factor criteria.

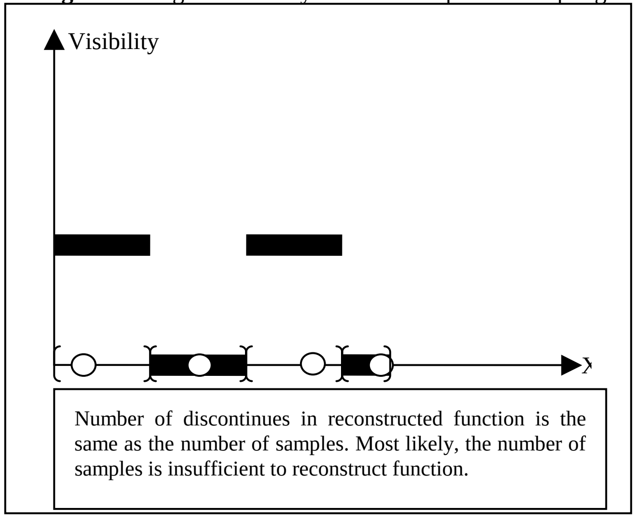
Experiments show that pivot point approach performs very well when points in the packet lie in the same plane or ‘see’ approximately the same portion of the light source. Otherwise it is better to split packet and process each point individually.

7. NOISE MEASURE



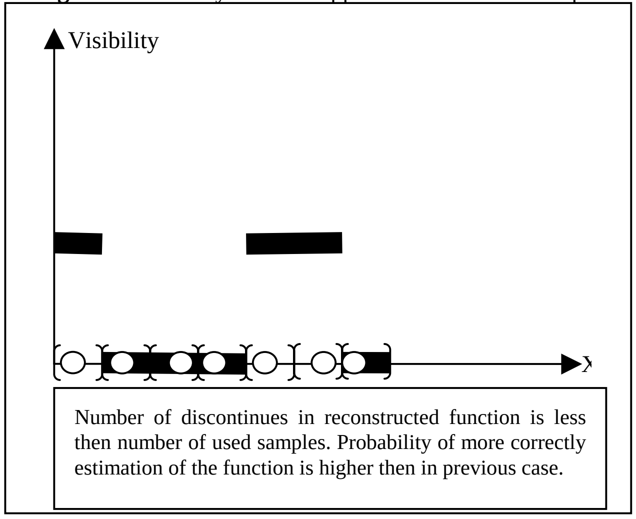
Ideal samples distribution. Each sample is located in the interval of visibility function continuity and vice versa. Number of discontinues in function reconstructed from the samples is the same as the number of discontinues in original function.

Figure 1: Original visibility function and perfect sampling.



Number of discontinues in reconstructed function is the same as the number of samples. Most likely, the number of samples is insufficient to reconstruct function.

Figure 2: Visibility function approximation with 4 samples.



Number of discontinues in reconstructed function is less then number of used samples. Probability of more correctly estimation of the function is higher then in previous case.

Figure 3: Visibility function approximation with 7 samples.

As a matter of fact, procedure of visibility estimation is an integration of discontinuous function. In such case discontinuities are the main source of errors or artifacts. So if we could better approximate locations of discontinuous and number of discontinues we could achieve better results. In case of using stochastic sampling number of discontinuities of visibility function is the most important characteristic of noise prediction. For achieving high image quality, noise elimination is critical for us. So this immediately raises the question of simple and adequate measure for noise as well as an algorithm to estimate it. One, and the most popular and simple, way is using information about neighborhood points. Another approach is using information only at specific point relying on samples only for this point. The latter one is more tricky and non-obvious, although, much more beneficial for exploiting various types of parallelism, because it process each point independently of others. Nevertheless our method allows easy extension to the case when neighborhood information can be used.

Assume we fixed a point in 3D space and estimate illumination for this point, then visibility is a function of points on light source. Let $V(x)$ is the true visibility at the fixed point. It is piecewise constant function with values 0 or 1 defined for $x \in S$, where S is a surface of the light source.

Assume that there exists unique subdivision of S : $S = \bigcup_{i=1..p} T_i$, so that $T_i \cap T_j = \varnothing$, each T_i is (linearly) connected set, and $\forall i, 1 \le i \le p$, $V(x) = V(y), \forall x, y \in T_i$ and p is minimal. In other words, each T_i is the maximum connected area where function $V(x)$ is constant, so subdivision is minimal in terms of number of T_i .

Numerically integrating function $V(x)$ over S , we perform a subdivision of S : $S = \bigcup_{i=1..m} S_i$, so that $S_i \cap S_j = \varnothing$, $\forall i \neq j$. We then randomly select one sample $y_i \in S_i$ at each S_i and compute visibility function value $V(y_i)$. So we reconstruct function $\tilde{V}(x) = \sum_{i=1...m} I(x \in S_i) V(y_i)$ which is an approximation for the $V(x)$ function from m samples. $I(x \in S_i)$ equals to 1 if x belongs to the S_i, and 0 – otherwise.

Finally, exact visibility $Visibility = \int V(x) \partial x$ will be approximated by $\tilde{Visibility} = \int \tilde{V}(x) \partial x$.

If we had perfect predictor we would be able to build such subdivision that $p = m$ and $S_i = T_i, \forall i$. But the probability of such case is negligible. More probably, we need more samples to make sure that we sufficiently approximated $\{T_i\}_{i=1..p}$ by $\{S_i\}_{i=1..m}$. But how much samples do we really need? Lets look at subdivision $\{S_i\}_{i=1..m}$. We can transform it by merging adjacent areas S_i and S_j , if $y_i = y_j$. So we get new subdivision $\left\{\tilde{S}_i\right\}_{i=1..v}$ of S . Essentially we can expect that if two adjacent areas S_i and S_j have the same visibility values ($y_i = y_j$) and if $\max_{i=1..m} \mu(S_i)$ is small enough then such areas belong to the same T_r , for some r . If v is close to m then we probably need using more samples to be sure that we correctly approximated $\{T_i\}_{i=1..p}$ by $\left\{\tilde{S}_i\right\}_{i=1..v}$. If v is much smaller than m then we can be sure that we approximated $\{T_i\}_{i=1..p}$ sufficiently. So we can expect that ratio $\frac{v}{m}$ is a good estimation of measure of noise. Grater values of the ratio correspond to higher noise and smaller ones – to lower noise.

As a result – we integrate discontinuous function (visibility function). We use some number of samples to do this. So, more precisely, we reconstruct discontinuous function by some number of samples. Reconstructed function has some number of discontinuities. Our measure for noise is ratio of number of discontinuities of reconstructed function to number samples used to reconstruct this function.

Figure 4 illustrates visual dependency between our noise level estimation and visual quality of the image. Note that additional computations were performed executed only where our noise measure was too high.

8. BOUNDARY-VALUE CALCULATION OF ILLUMINATION

In spite of the fact that we have tool to make noise level uniform over image but due to low samples budget we still get grainy image and it might take tons of computations to eliminate noise completely. It is possible to perform some intelligent filtering to remove noise and keep sharp edges at the same time. We would like to avoid filtering step, though, for the following reasons:

1. Good filtering requires storing additional data together with the frame buffer (including such global information as normal, id of triangle etc.). Such data is required for each bounce of reflection/refraction and effectively kills any efforts of efficient usage of caches and causes tremendous traffic to external memory.
2. Filtering over image becomes the additional stage of the synchronization in parallel implementation which is critical for animation speed.

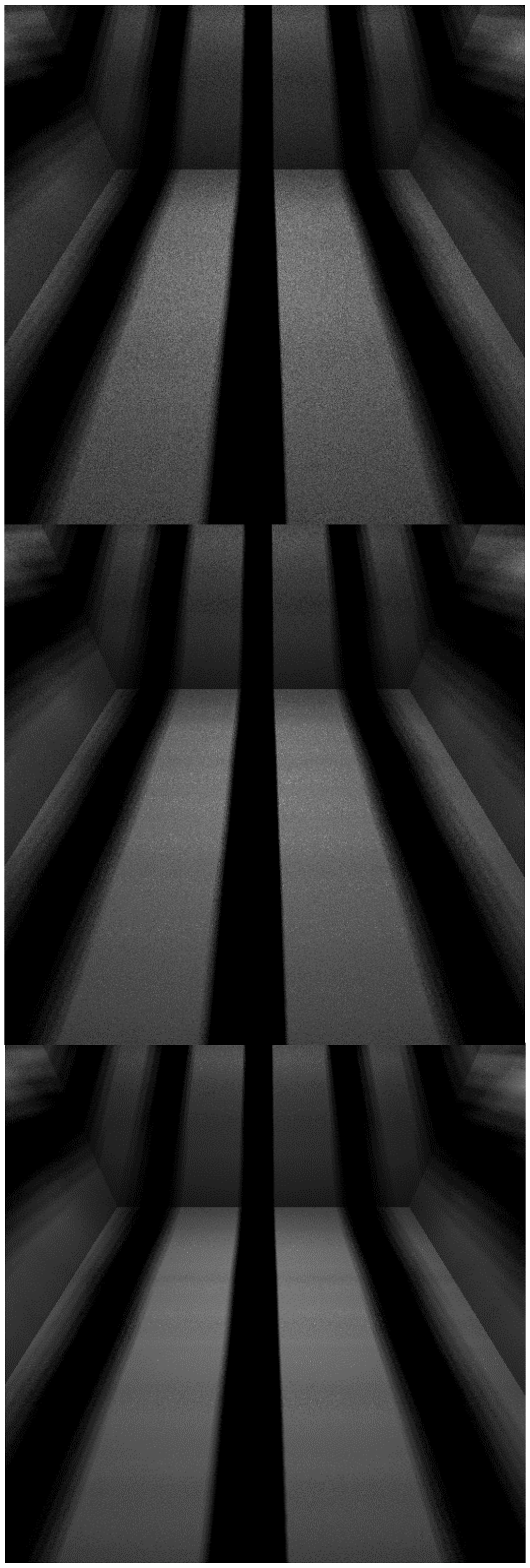
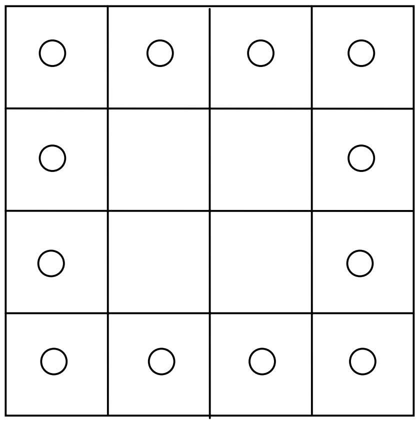


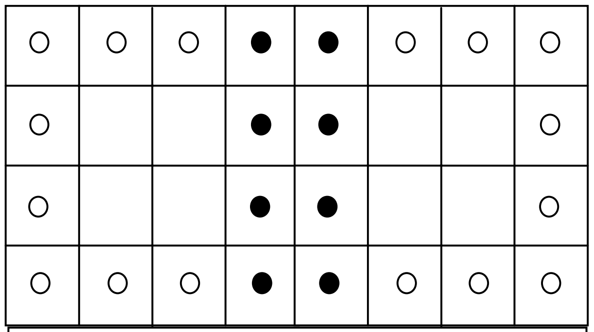
Figure 4: Noise measure (top-down): 0.95, 0.3, 0.2.

We suggest the following way to solve the problem of noise filtering. Fortunately, our illumination computation algorithm receives some group of points for processing. Usually this group of points represents hit points of rays originating from group of 4x4 pixels (boundary pixels marked by white points):



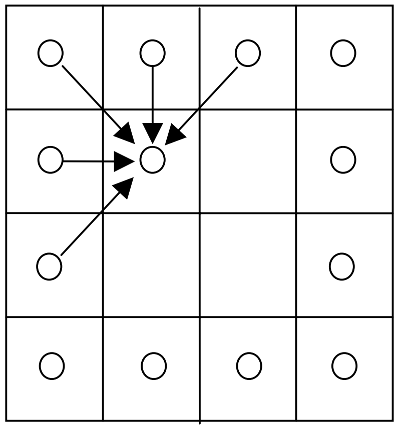
For the boundary points of the group we are computing illumination as precise as possible.

Goal then is to make sure that illumination changes slowly from one boundary to another for adjacent groups of pixels if it is changing slowly in the scene:



Illumination at black colored points must have a little difference if illumination in the scene changes slowly.

Next step is to compute illumination for the inner points of the group by using already computed values for neighboring points. It is a boundary-value problem. We use weighted sum to solve it:



Boundary-only illumination evaluation gives 1.3 performance gains in case of 4x4 point group while preserving the same image quality and producing slightly lower noise (see **Figure 5**, **6**).

Note that boundary-only algorithm allows spending saved computation budget to achieve better results than per-pixel algorithm.

Also note that approach does not prevent one from filtering illumination values on the boundary before evaluation of inner illuminations. This can be done to eliminate high-frequency noise.

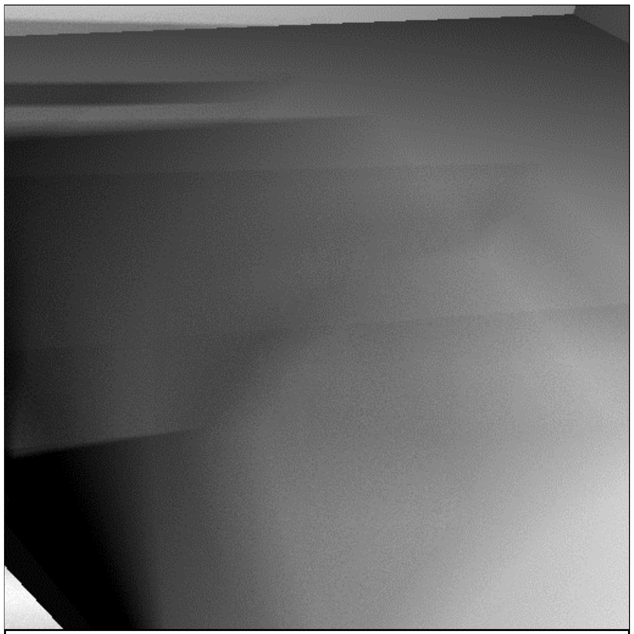


Figure 5: per-pixel illumination evaluation

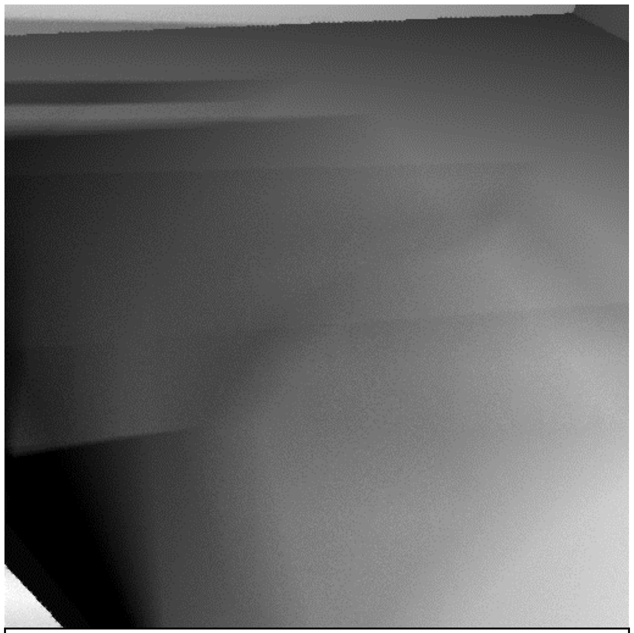


Figure 6: boundary-only illumination evaluation

9. ALGORITHM

Our algorithm has two parameters. First one is called ‘illumination threshold’ and responsible for the initial subdivision of light source. Second one is called ‘noise threshold’ and drives subdivision. First, light source is subdivided into patches according to the illumination threshold in such a way, that each patch has equal form-factor relatively to the specific point. Then fixed number of visibility tests for each 4x4 point group is performed and the estimator is deployed to choose among one of three options:

1. If all samples on a patch are visible (occluded) then assume that patch is fully visible (fully occluded). So compute illumination from this patch, compute intensity value and add it to the total value for the point.
2. If patch is partially visible (some samples are visible and some are occluded) and the noise measure is higher than defined noise threshold then subdivide the patch into sub-patches, add them to patch queue and proceed to the next patch.
3. If patch is partially visible and noise measure is lower then the threshold then compute intensity and add its value to the total value (using illumination from the patch multiplied by visibility estimated from samples). Proceed to the next patch.

10. CONCLUSION

We presented an approach for photo-realistic soft shadows rendering that has the following properties.

1. Requires on average only 20% of visibility tests brute-force approach uses to achieve the same image quality.
2. Uses proposed simple noise measure allowing adaptively distribute available visibility tests budget taking more samples only where it’s really needed.
3. Is straightforwardly parallelizable.
4. Applicable for and efficiently uses tracing packet of multiple rays simultaneously.

11. FUTURE WORK

Presented method works very well with one or some small number of area/linear lights, it would be interesting to come up with extension for large number of lights.

Although the algorithm does not require specification of absolute value for intensity threshold (which makes it suitable for using any tone mapping on the post processing stage) there are still two threshold parameters that need to be specified at the input. So investigation of ways to determine them automatically is another future topic.

Since our ray tracing core is very fast, another potential extension is hybrid method which using knowledge of performance of visibility tests can automatically switch to brute force algorithm in the areas where visibility tests are inexpensive.

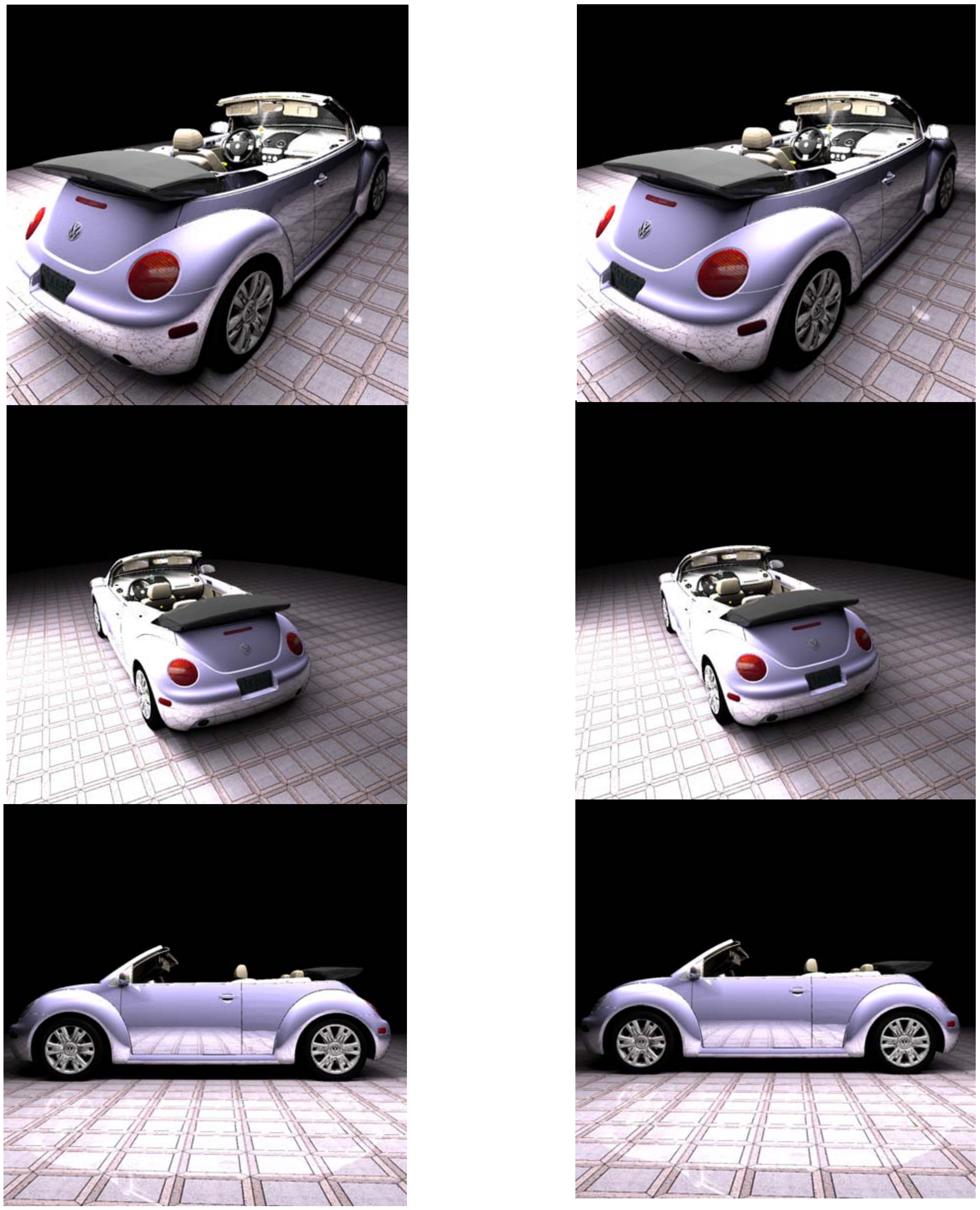


Figure 7: VW Beetle model (2M triangles). All images were generated at resolution 1024x1024 and have one reflection level. Left column – images, generated by our method, right – images, generated by brute-force method. Brute-force method used 100 samples per pixel. Our method required on average 5 times as less samples as the brute force method. Execution times for left column (top-down): 10.78 sec, 13.08 sec, 14.86 sec. Times for execution for right column (top-down): 25.23 sec, 35.20 sec, and 37.45 sec.

12. REFERENCES

[1] L´aszl´o Szirmay-Kalos, Werner Purgathofer. *Analysis of the Quasi-Monte Carlo Integration of the Rendering Equation.* Proc. 7th Int. Conf. in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG 1999), Univ. of West Bohemia Press, 1999.

[2] Tomoyuki Nishita, Isao Okamura, Iyachiro Nakamae. *Shading Models for Point and Linear Sources*. ACM Transactions on Graphics (TOG). Volume 4, Issue 2 (April 1985). Pages: 124 – 146.

[3] Marc J. Ouellette, Eugene Fiume. *On Numerical Solutions to One-Dimensional Integration Problems with Applications to Linear Light Sources*. ACM Transactions on Graphics (TOG). Volume 20, Issue 4 (October 2001). Pages: 232 – 279.

[4] Peter Shirley, Changyaw Wang, Kurt Zimmerman. *Monte Carlo Techniques for Direct Lighting Calculations*. ACM Transactions on Graphics (TOG). Volume 15, Issue 1 (January 1996). Pages: 1 – 36.

[5] Andrew Gardner, Chris Tchou, Tim Hawkins, Paul Debevec. *Linear Light Source Reflectometry*. ACM Transactions on Graphics (TOG). Volume 22, Issue 3 (July 2003). Special issue: Proceedings of ACM SIGGRAPH 2003. Session: Scattering and reflectance measurement. Pages: 749 – 758.

[6] Norman Chin, Steven Weiner. *Fast Object-Precision Shadow Generation for Area Light Sources using BSP Trees*. Symposium on Interactive 3D Graphics. Proceedings of the 1992 symposium on Interactive 3D graphics. Pages: 21–30.

About the authors

Moiseenko, Yuri is a research scientist at Architecture Research Lab, Corporate Technology Group, Intel Corp.

His contact email is Yuri.Moiseenko@intel.com.

Sevastianov, Igor is a research scientist at Architecture Research Lab, Corporate Technology Group, Intel Corp.

His contact email is Igor.Sevastianov@intel.com.

Soupikov, Alexei is a research scientist at Architecture Research Lab, Corporate Technology Group, Intel Corp.

His contact email is Alexei.Soupikov@intel.com.