



Constant-Memory Order-Independent Transparency Techniques

Louis Bavoil

lbavoil@nvidia.com

Eric Enderton

eenderton@nvidia.com

March 2011

Document Change History

Version	Date	Responsible	Reason for Change
1	March 14, 2011	Louis Bavoil	Initial release

Overview

Order-Independent Transparency (OIT) algorithms are algorithms for which the output image is independent on the order of the input primitives. This SDK sample includes DirectX 11 implementations of Stochastic Transparency [Enderton et al. 2010] and Dual Depth Peeling [Bavoil and Myers 2008]. It uses the "independent blend" hardware feature for Dual Depth Peeling and the output coverage mask feature for Stochastic Transparency.

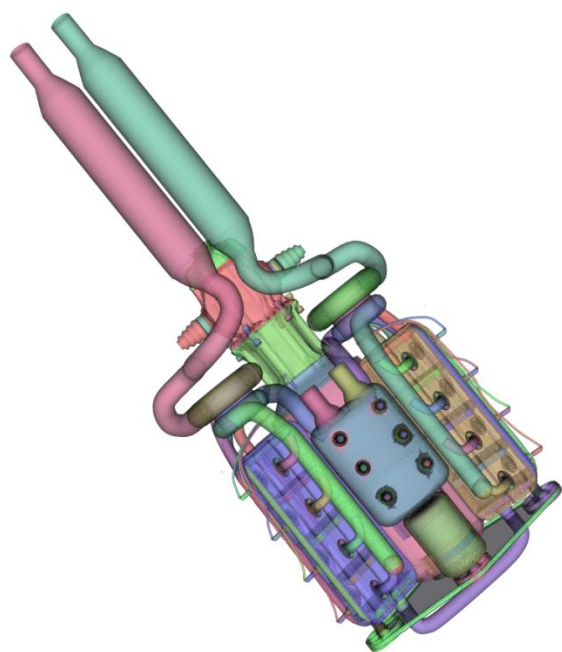


Figure 1. Full stochastic transparency algorithm [Enderton et al. 2010] with 3 geometry passes and 8x MSAA.



nvidia.

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com

Introduction

Depth peeling [Mammen 1989] [Everitt 2001] is a robust and simple OIT algorithm. The algorithm requires performing N geometry passes for peeling N color layers. Dual depth peeling [Bavoil and Myers 2008] is an accelerated depth peeling algorithm, which can peel N layers in $N/2+1$ geometry passes. Both depth peeling and dual-depth peeling can be implemented with a constant video-memory budget, by blending the layers on the fly as they get captured in front-to-back or back-to-front order. These algorithms require a view-dependent number of geometry passes to converge to the correct result.

A different approach is to capture all the semi-transparent fragments first, before sorting and blending them. This is what the original A-Buffer algorithm does to handle transparency [Carpenter 1984]. Using DirectX 11 GPUs, it is now possible to capture all the transparent fragments using per-pixel linked lists in a single geometry pass [Yang et al. 2010]. The fragments are then sorted and blended in a post-processing pass. The main issue with this algorithm is that it requires a dynamic amount of video memory to store all the visible semi-transparent fragments, which may be impractical for some applications where the number of semi-transparent fragments is unbounded.

Bucket depth peeling [Liu et al. 2009] and Stochastic Transparency [Enderton et al. 2010] are approximate OIT algorithms that can produce plausible results with a constant video-memory budget. Another constant-memory OIT algorithm intended mostly for hair was proposed by [Sintorn and Assarson 2009]. An advantage of the Stochastic Transparency algorithm is that it always converges to the correct result when increasing the effective number of samples per pixel, without increasing the video-memory budget.

Implementation Notes

Stochastic Transparency

For Stochastic Transparency, each geometry pass uses 8 MSAA samples per pixel and multiple passes can be performed to simulate more MSAA samples per pixel. Alpha correction is enabled by default and can be disabled by setting `USE_ALPHA_CORRECTION=0` in `StochasticTransparency11.hlsl`.

The implementation uses the `RGBA16F` format for the MSAA color buffer (see `STOCHASTIC_COLOR_FORMAT` in `StochasticTransparency11.h`). Using a `RGBA8` format introduces a significant bias in the results if `USE_ALPHA_CORRECTION=1`.

Running the Sample

The model can be rotated by dragging the mouse on the window. The following rendering algorithms are exposed in the GUI: stochastic transparency, dual depth peeling, and plain alpha blending (no OIT). The model is rendered with a uniform opacity, which is exposed in the GUI with the “Alpha” slider.

Results

Figure 2 compares Stochastic Transparency with Dual Depth Peeling for similar number of geometry passes. Stochastic Transparency has the advantage of being naturally anti-aliased. In this implementation, it has 8xMSAA quality because it uses 8 samples per pixel for its MSAA buffers.

With the motor scene (322,284 triangles) in 1920x1200 on GeForce GTX 580, the frame times are 5.2 ms for Stochastic Transparency and 5.5 ms for Dual Depth Peeling, using 3 geometry passes for Stochastic Transparency and 4 for Dual Depth Peeling.

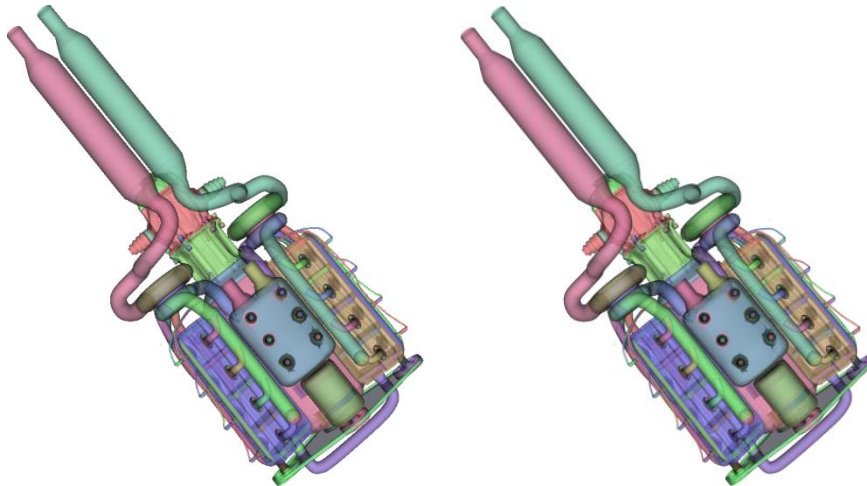


Figure 2. (left) Dual Depth Peeling with 4 geometry passes.
(right) Stochastic Transparency with 3 geometry passes.

References

- [Enderton et al. 2010] Eric Enderton, Erik Sintorn, Peter Shirley, David Luebke, “Stochastic Transparency”, Symposium on Interactive 3D Graphics and Games (I3D) 2010.
- [Yang et al. 2010] Yang, J. C., Hensley, J., Grün, H. and Thibieroz, N., “Real-Time Concurrent Linked List Construction on the GPU”, Computer Graphics Forum, 29, 2010.
- [Liu et al. 2009] Fang Liu, Meng-Cheng Huang, Xue-Hui Liu, En-Hua Wu, “Efficient Depth Peeling via Bucket Sort”, HPG '09 Proceedings of the Conference on High Performance Graphics 2009.
- [Sintorn and Assarson 2009] Erik Sintorn, Ulf Assarson, “Hair Self Shadowing and Transparency Depth Ordering Using Occupancy maps”, Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D 2009).
- [Bavoil and Myers 2008] Louis Bavoil and Kevin Myers, "Order Independent Transparency with Dual Depth Peeling", NVIDIA, 2008.
- [Everitt 2001] Cass Everitt, “Interactive Order-Independent Transparency”, NVIDIA, 2001.
- [Mammen 1989] Abraham Mammen, “Transparency and Antialiasing Algorithms Implemented with the Virtual Pixel Maps Technique”, IEEE Computer Graphics and Applications, v.9 n.4, 1989.
- [Carpenter 1984] Loren Carpenter, “The A-Buffer, an Antialiased Hidden Surface Method”, ACM SIGGRAPH Computer Graphics, v.18 n.3, p.103-108, 1984.

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, GeForce, NVIDIA Quadro, and NVIDIA CUDA are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2011 NVIDIA Corporation. All rights reserved.