



Stencil Routed K-Buffer

Kevin Myers
kmyers@nvidia.com

Louis Bavoil
lbavoil@nvidia.com

November 2007

Document Change History

Version	Date	Responsible	Reason for Change
1.0	November 6, 2007	Louis Bavoil	Initial release

Overview

GPUs are designed to handle traditional rasterization, where only the nearest fragment is kept (Z-buffering). This sample shows how to use stencil routing to capture 8 layers of fragments per pixel per geometry pass. First, up to 16 fragments per pixel are captured in rasterization order, in 2 geometry passes. Second, a fullscreen shader pass sorts the 16 fragments per pixel using a bitonic sort. Finally, another fullscreen shader pass renders either a translucency effect based on volumetric absorption and a dual-layer refraction approximation (Figure 1), order independent alpha blending, or the individual layers.

To compare performance, the sample also features a depth peeling mode, which captures one layer per geometry pass. The advantage of depth peeling is that it captures its layers in depth order. However, stencil routing is faster when it costs more to render the geometry than to sort, and when all the fragments need to be captured.



Figure 1. Example of volumetric effect rendered with a stencil routed k-buffer with 16 fragments per pixel.



NVIDIA.

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

Implementation Details

The code is organized as follows. The `StencilRoutedKBuffer` and `DepthPeeledKBuffer` classes are defined in `KBuffer.h`. They implement the `Capture` and `Postprocess` methods of the `KBuffer` abstract class. In addition, `StencilRoutedKBuffer` also provides a method to query the number of overflowed pixels (rendering a fullscreen quad with an occlusion query), and to asynchronously read the number of overflows.

Running the Sample

This sample requires DirectX 10. Because the algorithm relies on the ability to fetch the individual samples of a multisample texture, it cannot be implemented with DirectX 9.

Use the mouse to rotate the camera, and the A/D/S/W keys to pan.

Limitations

On the GeForce 8 series, up to 8 fragments per pixel must be computed in one pass over the geometry. To capture more layers, more geometry passes are required.

Because the stencil routed k-buffer algorithm stores the fragments in a multisample texture, it requires multisample antialiasing (MSAA) to be disabled when rendering to the k-buffer.

References

"Stencil Routed A-Buffer", Kevin Myers, Louis Bavoil, ACM SIGGRAPH 2007 Technical Sketch Program. <http://developer.nvidia.com/object/siggraph-2007.html>

"Deferred Rendering using a Stencil Routed K-Buffer", Louis Bavoil, Kevin Myers, ShaderX 6 - Advanced Rendering Techniques (to appear).

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

© 2007 NVIDIA Corporation. All rights reserved.