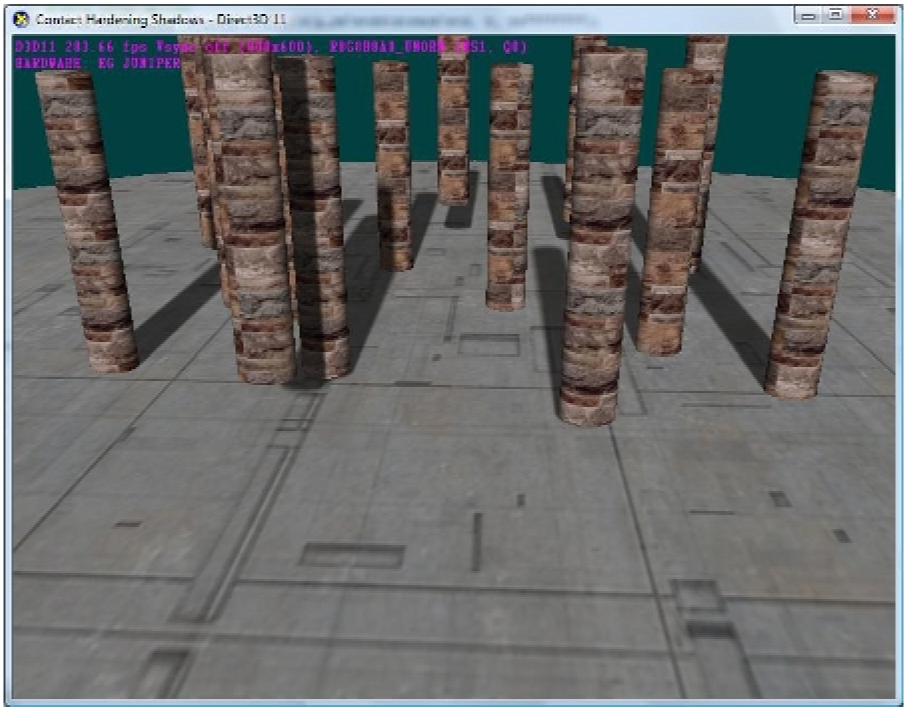
ContactHardeningShadows11 Sample

⊟ Collapse All

This sample, contributed by AMD, presents a Direct3D 11 / Shader Model 5.0 accelerated technique that uses non-stationary / dynamic weight matrices for unique per-sample filter weights (these don't necessarily represent a separable filter) for a contact hardening shadow effect. The effect is comparable to PCSS (Percentage-Closer Soft Shadows [Fer05]) but does not suffer from banding or noise artifacts.



```
D3D11 283.66 fps Vsync off (800x600), R8G8B8A8_UNORM (MS1, Q0)
HARDWARE: EG JUNIPER
```

Path

Source	SDK root\Samples\C++\Direct3D11\ContactHardeningShadows11
Executable	SDK root\Samples\C++\Direct3D11\Bin\x86 or x64\ContactHardeningShadows11.exe

Introduction

Most games make use of uniform shadow filters that don't produce physically plausible shadows. This means, for example, that the blurriness of shadows does not depend on the distance between shadow caster and shadow receiver.

In order to achieve physically plausible shadows, PCSS (Percentage Closer Soft Shadows [Fer05]) were introduced as a real-time method to achieve contact hardening shadows. Typical implementations of PCSS suffer from noise and banding artifacts that result from the use of a Poisson disk of samples.

This SDK sample uses a large box (12×12) of shadow map samples in combination with a non-stationary filter weights matrix to achieve a transition from sharp to soft shadows. The sample accepts the fact that one needs to limit the size of the light source in order to achieve high quality results. Since the sun can usually be treated as a relatively small light source, the technique works well for a directional light.

The Shader Model 5.0 instructions GatherRed()and GatherCmpRed()are used to accelerate the computation of average-blocker-depth and to accelerate the non-stationary and non-separable dynamic filter operation.

Blocker Search

The search for blockers for a N×N shadow map filter footprint is carried out using only (N/2)×(N/2) GatherRed() operations. The shader in the sample performs 36 of these instructions. A Shader Model 4.0 implementation would need to perform 144 point samples in order to obtain the same information.

Filtering with a Dynamic Filter Matrix

Based on the average blocker depth and the size of the light source, a factor between 0.0 (sharp) to 1.0 (totally blurry) is computed. This factor is used to compute a dynamic weight matrix that results from feeding four matrices into a cubic Bezier function. The math from [GS09] is used to compute the filter, reducing the necessary ALU through the use of GatherCmpRed().

References

[GS09] Gruen Holger, Story Jon, "Taking Advantage of Direct3D 10.1 to Accelerate Performance and Enhance Quality," AMD sponsored session at Eurographics 09, available online at http://developer.amd.com/gpu_assets/TakingAdvantageofDirect3D10.pps

[Fer05] Fernando Randima, ”Percentage-closer soft shadows”, July 2005, SIGGRAPH '05: SIGGRAPH 2005 Sketches

Send feedback to DxSdkDoc@microsoft.com.
Version: 1962.00