
GDC
09
learn
network
inspire
www.GDConf.com
Game Developers Conference®
March 23-27, 2009 | Moscone Center, San Francisco

High Quality Direct3D 10.0 & 10.1 Accelerated Techniques

Jon Story, AMD
Holger Gruen, AMD

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

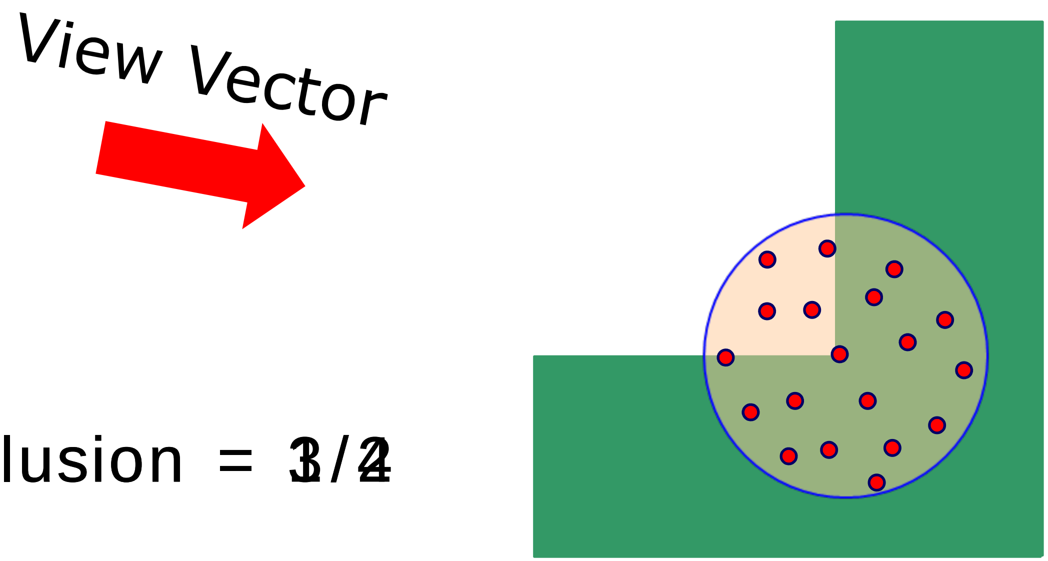
Agenda

- High Definition Ambient Occlusion
- High Quality Shadow Filtering

High Definition Ambient Occlusion (HDAO)

Conventional AO : 1

- » [illegible]
- » Occlusion Factor ~= Failure Rate



Occlusion = 3/4

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Conventional AO : 2

- Usually requires many depth samples to achieve acceptable results
- Maths overhead per sample is high
  - Transform to post-perspective space
  - Depth testing
  - Result attenuation
- Filtering pass almost essential to smooth out dithering and banding artifacts

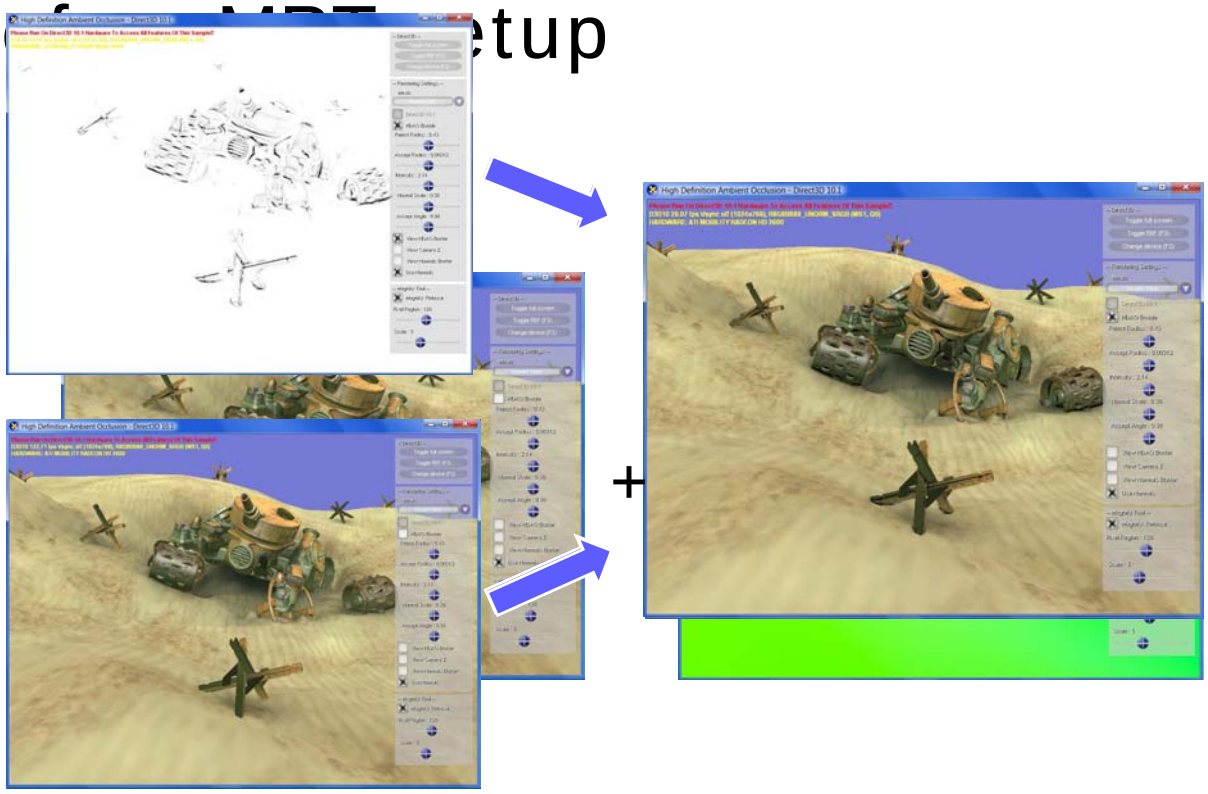
THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

The Aim of HDAO

- Deliver a believable AO look
- Achieve affordable performance on today‘s HW
- Avoid need for filtering pass(es)
  - Keep performance higher
  - No additional render targets required
- Easy to incorporate
  - No normals required
  - Account for normals with ease if available

How does HDAO Fit into the Rendering Pipeline?

- Render scene (and normals)
- Optionally render normals as part of MRT setup

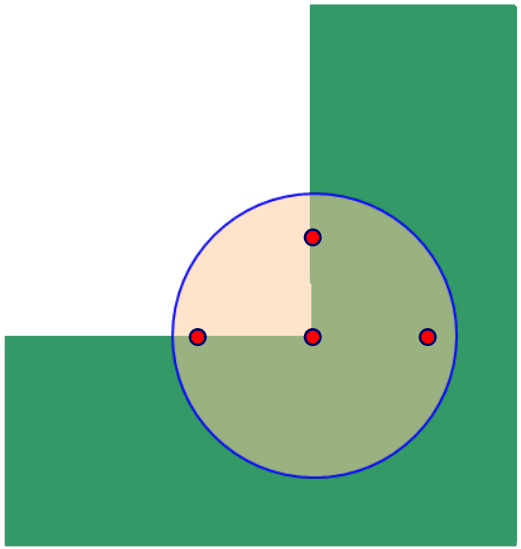


How does HDAO Work?

- Detects valleys in camera Z space
- Not _only_ creases ☺

View Vector

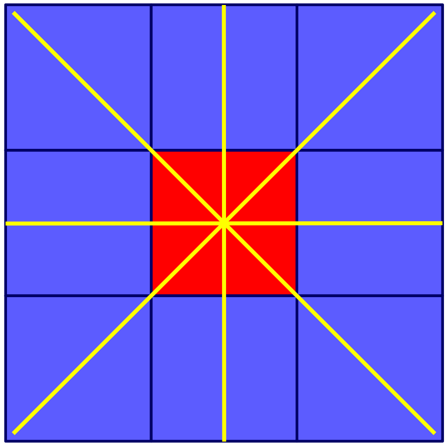
Valley = false



THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Valley Detection

- Start by sampling the central depth (raw Z) converted to camera Z
- The simplest implementation would be a 3x3 sample pattern
- Perform valley detection for each twin point (pixel pair) comparison
- Valley test produces binary result



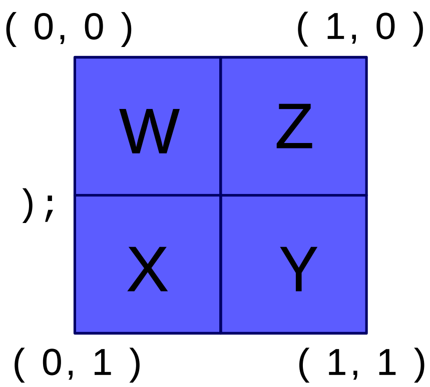
Credit to Holger Gruen for this idea

Occlusion Factor = Valley Test

How does Gather Work?

- Gather fetches 4 point sampled texels in a single instruction
- Similar to Fetch4 exposed on DX9 ATI GPUs
- Available in Direct3D 10.1 & 11 hardware
- There are known issues in previous versions of the HLSL compiler
- Used on single channel formats (restriction gone for DX11)
  - depth buffer
  - Shadow maps
- Must use March 2009 SDK or later

```
f4Depth = DepthTex.Gather( SamPoint, f2Coord );
```



THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Direct3D 10.0 Version of Gather

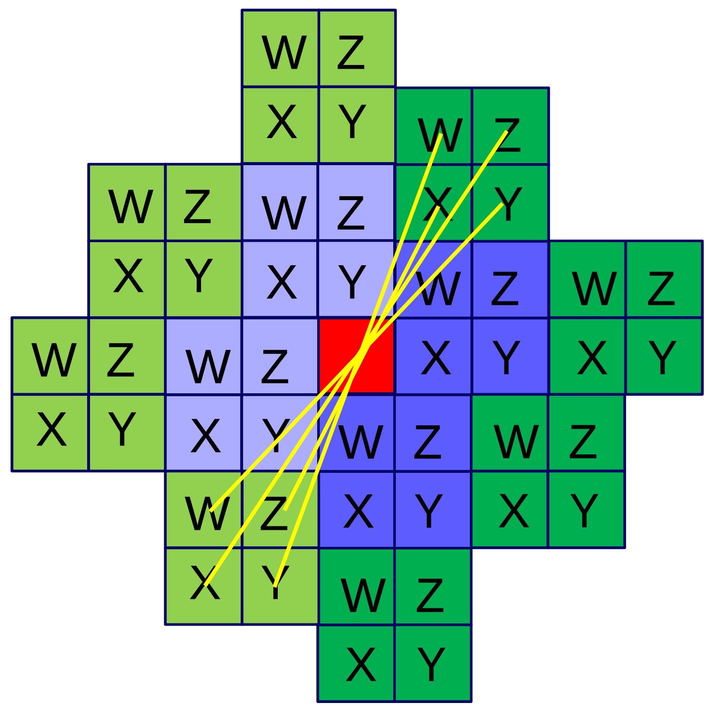
- Be sure to get the integer offsets correct for the 4 samples

```
// Direct3D 10.1
f4Ret = Tex.Gather( g_SamplePoint, f2TexCoord );

// Direct3D 10.0
f4Ret.x = Tex.SampleLevel( g_SamplePoint, f2TexCoord, 0, int2( 0, 1 ) ).x;
f4Ret.y = Tex.SampleLevel( g_SamplePoint, f2TexCoord, 0, int2( 1, 1 ) ).x;
f4Ret.z = Tex.SampleLevel( g_SamplePoint, f2TexCoord, 0, int2( 1, 0 ) ).x;
f4Ret.w = Tex.SampleLevel( g_SamplePoint, f2TexCoord, 0, int2( 0, 0 ) ).x;
```

Gather Pattern

- [illegible] only [illegible] depth sampling using Gather [illegible] swizzle xyzw with zwxy



Natural Vectorization

```
for( iGather=0; iGather<NUM_GATHERS; iGather++ )
{
    // Gather mirrored tw
    f4SampledZ[0] = Depth
    f4SampledZ[0] = -g_f
    f4SampledZ[1] = Depth
    f4SampledZ[1] = -g_fQ

    // Detect valleys
    // First twin
    f4Diff = fCenterZ.xxx
    f4Compare[0] = ( f4Di
    f4Compare[0] *= ( f4Di
    // Mirrored twin
    f4Diff = fCenterZ.xxxx - f4Sample
    f4Compare[1] = ( f4Diff < g_fHDA      dius.xxxx ) ? ( 1.0f ) : ( 0.0f );
    f4Compare[1] *= ( f4Diff > g_fHD      ceptRadius.xxxx ) ? ( 1.0f ) : ( 0.0f );
    // Accumulate occlusion factor
    f4Occlusion.xyzw += ( g_f4RingWeight[iGather].xyzw * f4Compare[0].xyzw *
    f4Compare[1].zwxy );
}
```

We perform the valley detection logic on 4 valleys at once

Finally we weight and store the occlusion factor of 4 valleys at a time

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

HDAO Buffer (depth only)

- 40 **Gathers**
- No filtering needed

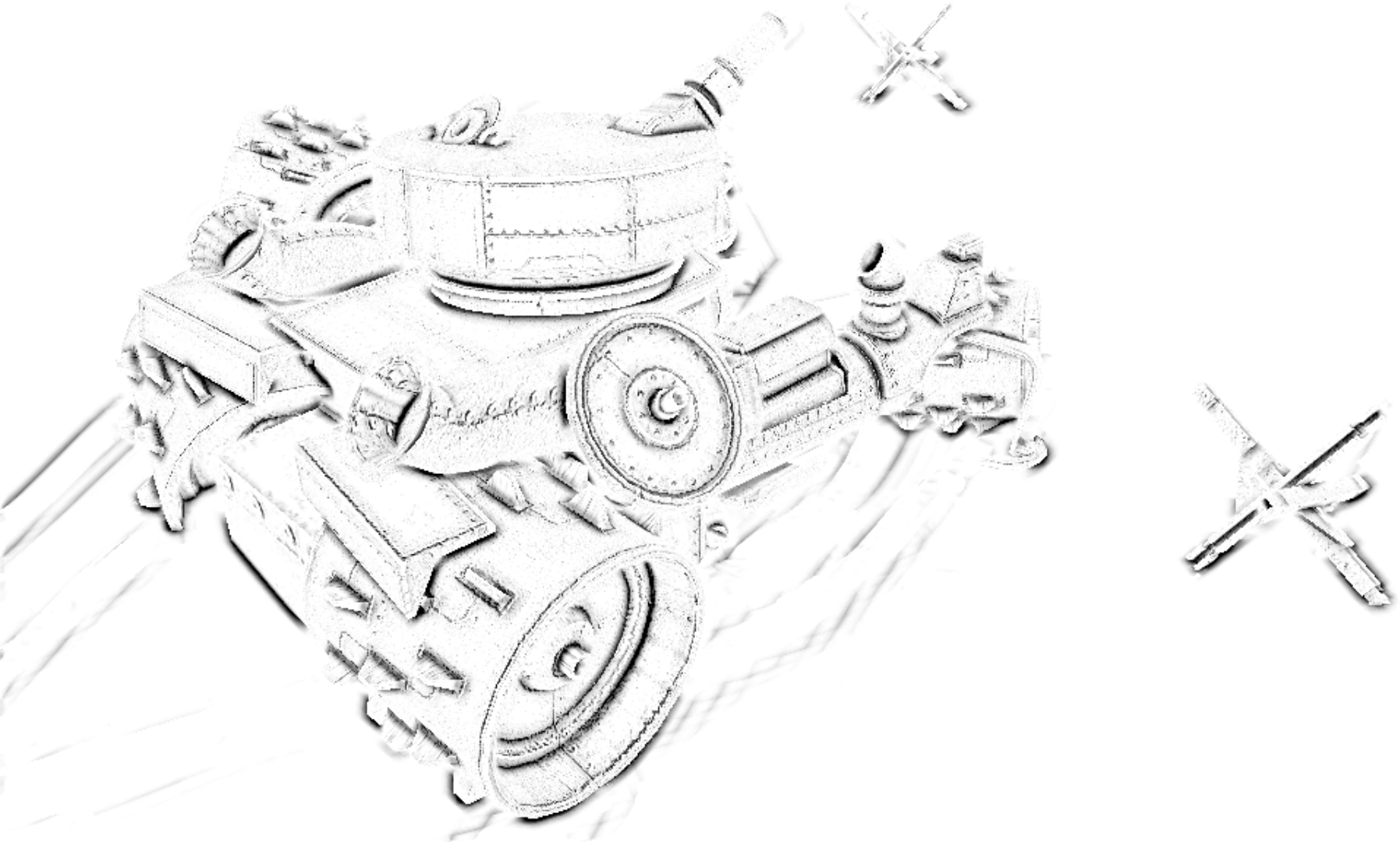
THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Bringing in Camera Space Normals

- HDAO easily accounts for normals
- Scale Z component of camera space normal by desired amount
- Add scaled normal to camera Z value
- Run valley detection code as before

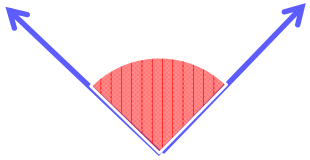
```
// Offset by scaled normal
f4CameraZ += ( f4NormalZ * g_fNormalScale );
```

HDAO - Buffer (depth & normals)
- 80 **Gathers** (could use alot less)
- No filtering needed

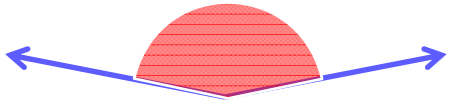


Early Rejection Test

- For mostly convex meshes, can produce good results
- Or compute direction vectors from full camera space positions
- Reject if too shallow



Pass

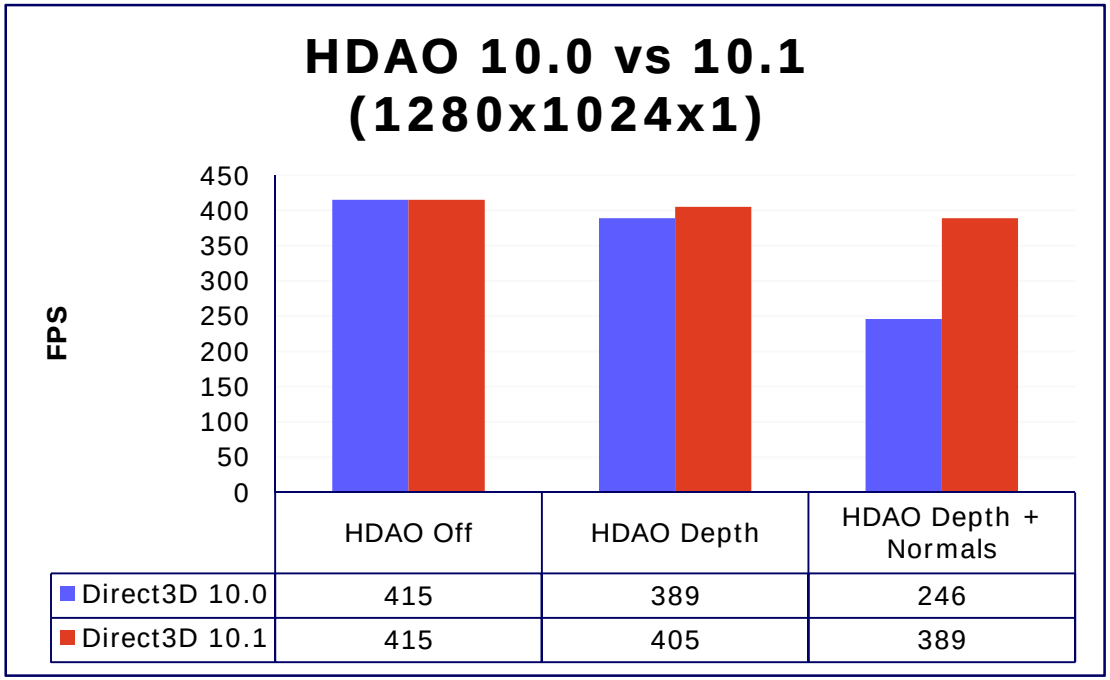


Fail

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Performance

- HDAO (depth and normals):
  - Direct3D 10.0: 0.65 MS
  - Direct3D 10.1: 0.59 MS



Phenom 2.3GHz, HD4870X2, 2 GB RAM, Windows Vista 32 (SP1)

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Tom Clancy‘s HAWX
Publisher: Ubisoft
Developer: Ubisoft Romania

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

THINK SERVICES
GDC 09
learn
network
inspire
www.GDConf.com

Stormrise
Publisher: SEGA
Developer: The Creative Assembly Australia

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

BattleForge
Publisher: EA
Developer: EA Phenomic

Future Work

- Looking into a compute shader accelerated version
  - Solid sampling pattern lends itself well to Thread Local Storage
- Account for strong light sources
  - AO for many scenes is not low frequency
- Real valley tracing...

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

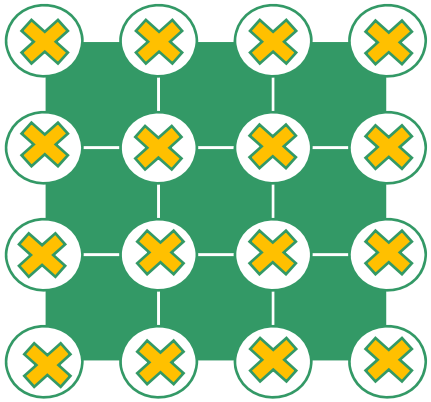
High Quality Shadow Filtering

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

How Direct3D 10.1 helps filtering

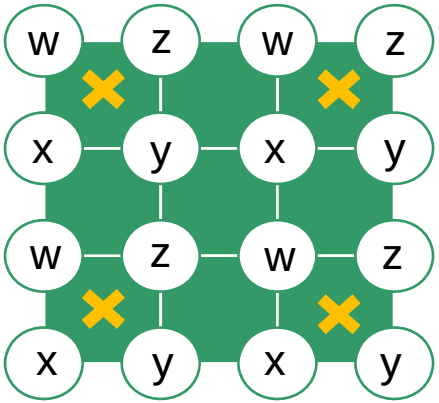
for single channel textures

Direct3D 10.0



NxN point samples if you need all data points – e.g. 4x4 = 16

Direct3D 10.1



(N/2)x(N/2) Gather operations get all data - e.g. 2x2 = 4

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Why revisit conventional shadow filtering? - 1

- There are advanced techniques for smooth shadows
  - The most prominent are
    - VSMs, layered VSMs, CSMs, ESMs, ACDF SMs
  - Can be combined with SATs for arbitary smoothness
- But these methods bring other problems
  - The renderer gets more complex
  - May need to work around specific artifacts
  - Use only if neccessary

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Why revisit conventional shadow filtering? - 2

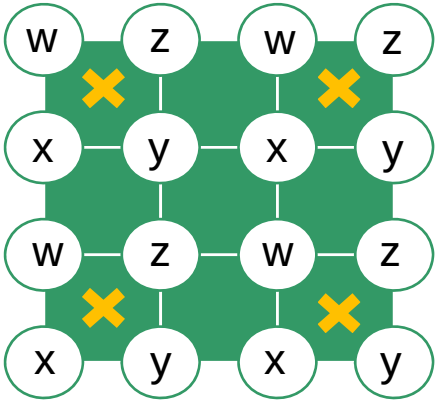
- Advanced methods come at a cost
  - More RTs at a high memory cost
  - Costly postprocessing operations
  - Non optimal RT formats
- Is an advanced techique needed?
  - Depth buffer based deferred shadowing does not depend on depth complexity
  - Big conventional shadow filters not that expensive

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Surprising insights about uniform shadow filtering 1

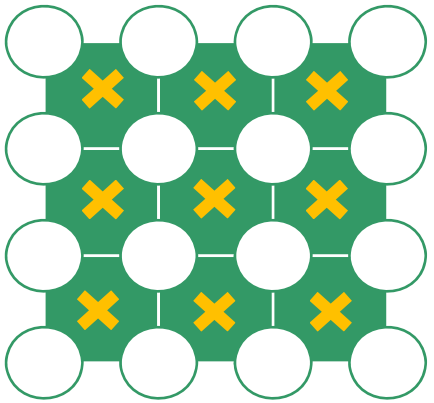
Let’s filter a 4x4 visibility sample block

Direct3D 10.1



4 Gather operations plus some ALU – (N/2)x(N/2) Gather ops for NxN

Direct3D 10.0



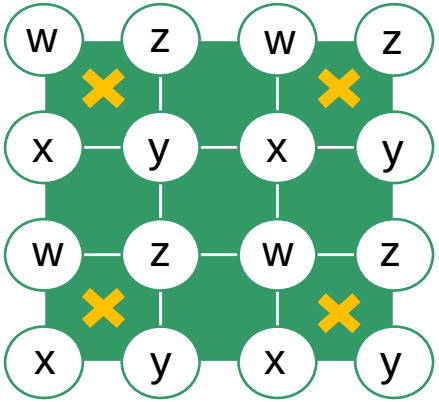
9 PCF samples plus some ALU right ?

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Surprising insights about uniform shadow filtering 1

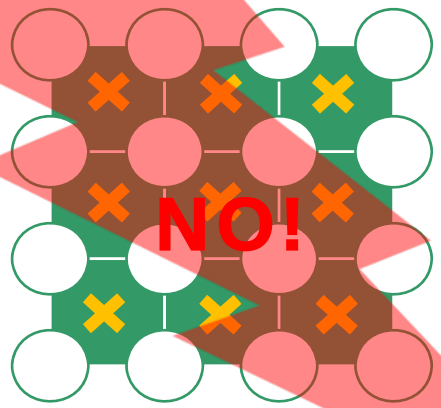
Let’s filter a 4x4 visibility sample block

Direct3D 10.1



4 Gather operations plus some ALU – (N/2)x(N/2) Gather ops for NxN

Direct3D 10.0



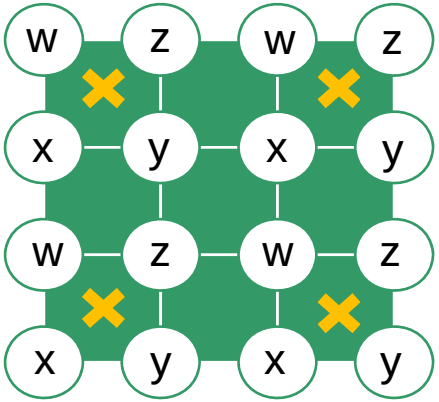
9 PCF samples plus some ALU right ?

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Surprising insights about uniform shadow filtering 1

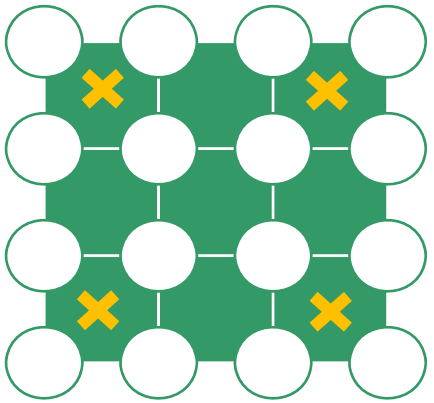
Let's filter a 4x4 visibility sample block

Direct3D 10.1



4 Gather operations plus some ALU => (N/2)x(N/2) Gather() samples for NxN

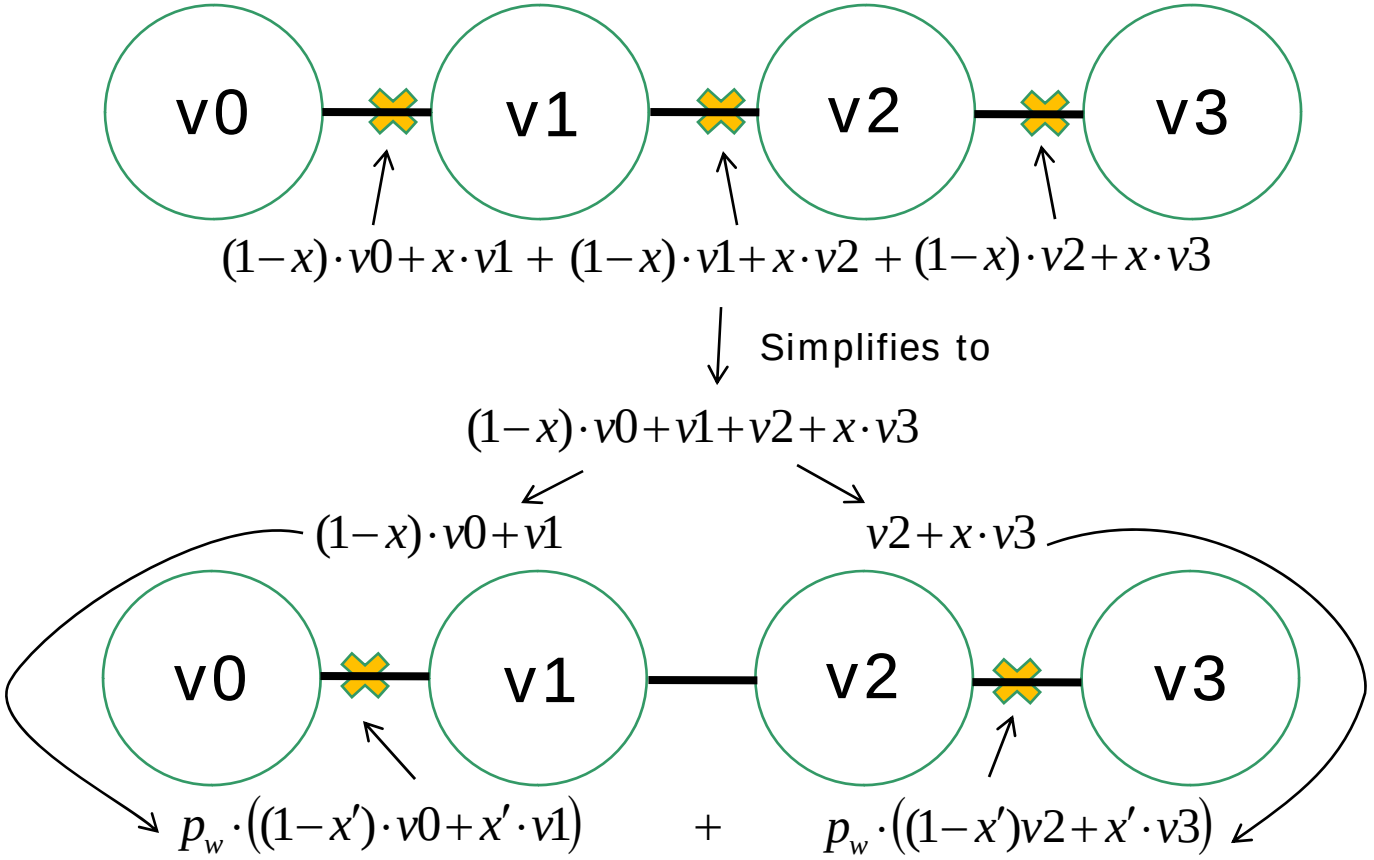
Direct3D 10.0



4 shifted PCF samples plus a post weight factor is enough => (N/2)x(N/2) PCF samples for NxN

Surprising insights about uniform shadow filtering 2

Let's look at only 1 row of 4 visibility samples



Credit for this idea goes to: Sergey Nenakhov at Funcom

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Surprising insights about uniform shadow filtering 3

- Only (N/2)x(N/2) PCF samples necessary instead for a uniform filter
- Cheaper than commonly assumed
  - 8x8 with only 16 PCF samples
  - Not only for shadow filtering
- Same texture op count as Direct3D 10.1
- Why bother with Direct3D 10.1?


From DICE's Frostbite Engine:
Uniform shadow filtering


From DICE's Frostbite Engine:
Gaussian shadow filtering

Disadvantages of uniform shadow filtering

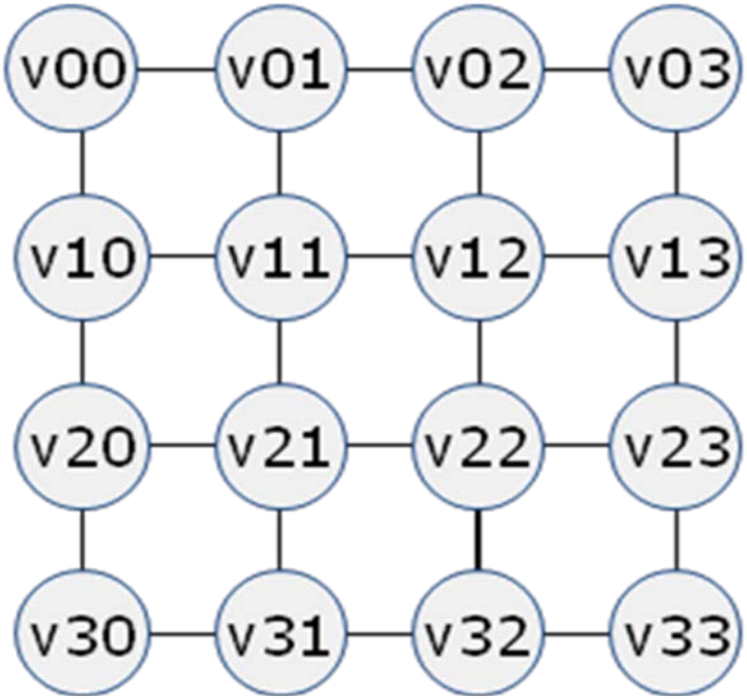
Uniform filtering blurs away too many details

Gaussian filtering preserves more details

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 1

Use a unique weight per PCF sample

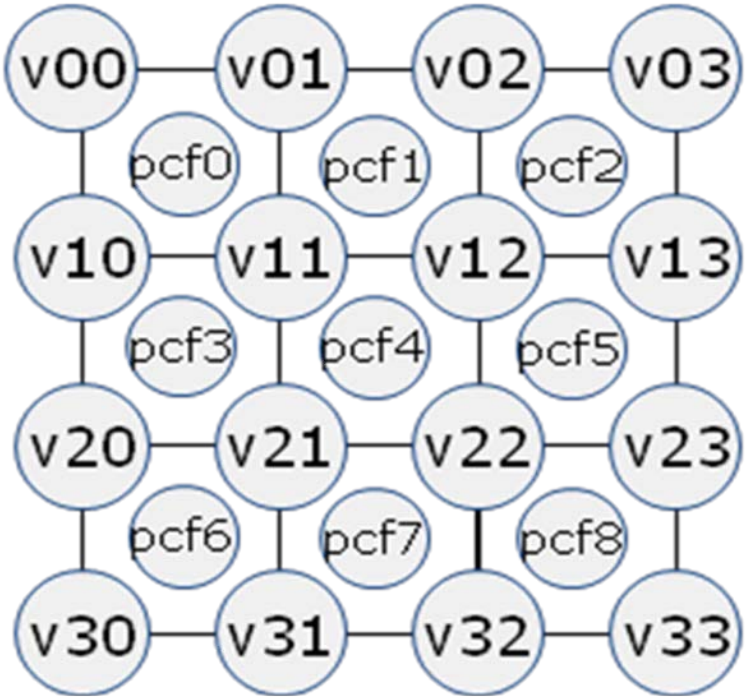


THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

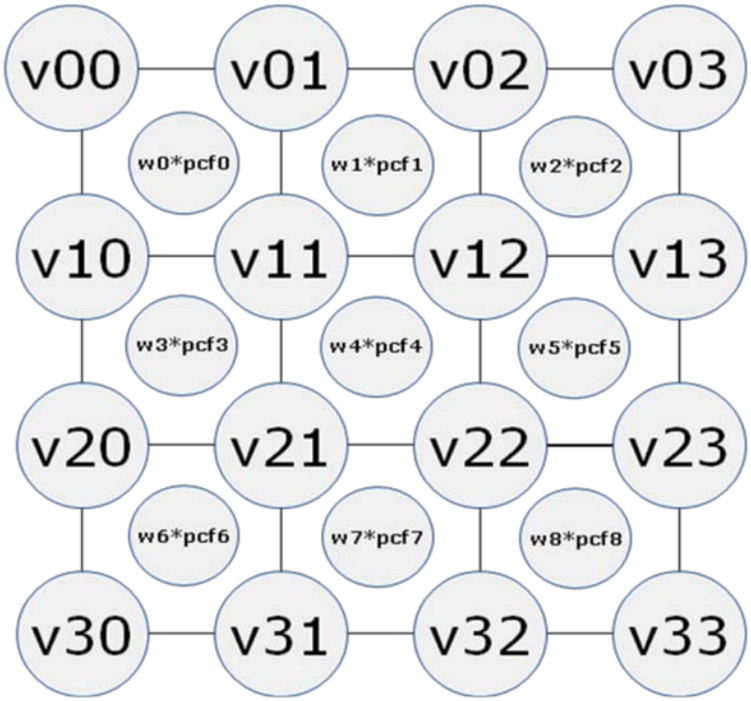
Advanced Direct3D 10.1 shadow filtering 1

Use a unique weight per PCF sample



Advanced Direct3D 10.1 shadow filtering 1

Use a unique weight per PCF sample



$$\frac{\sum_{k=0}^{(N-1)\cdot(N-1)} w_k \cdot pcf_k}{\sum_{k=0}^{(N-1)\cdot(N-1)} w_k}$$

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 2

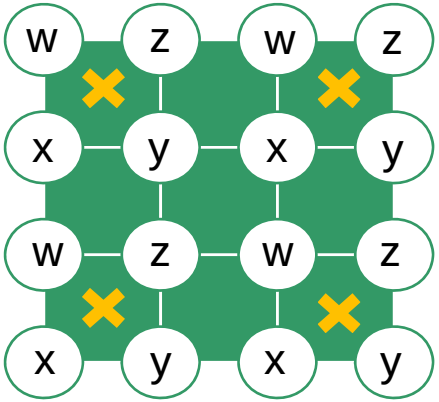
- Direct3D 10.1 needs (N/2)x(N/2) Gather() samples
- A (N/2)x(N/2) PCF samples solution is no longer possible for unique weights
  - Filter weights are not symmetric
    - Equation system not solvable
  - It is possible to get below NxN PCF ops for Direct3D 10.0 though

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 3

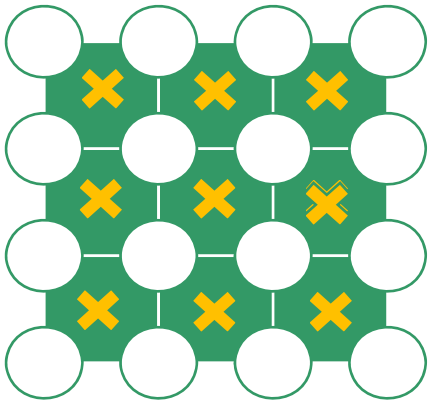
Let’s filter a 4x4 visibility sample block using unique weights

Direct3D 10.1



4 Gather() operations plus some ALU => (N/2)x(N/2) Gather samples for NxN

Direct3D 10.0



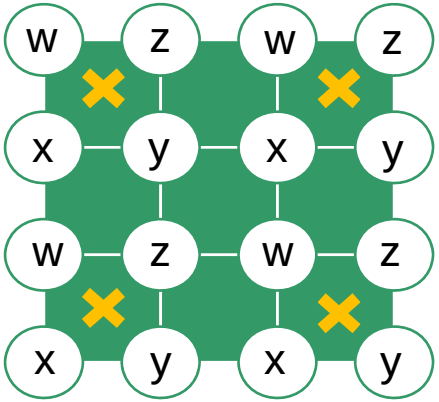
9 PCF samples plus some ALU right ?

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 3

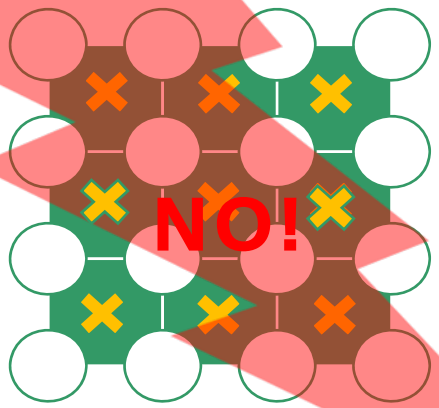
Let’s filter a 4x4 visibility sample block using unique weights

Direct3D 10.1



4 Gather() samples plus some ALU => (N/2)x(N/2) Gather() samples for NxN

Direct3D 10.0



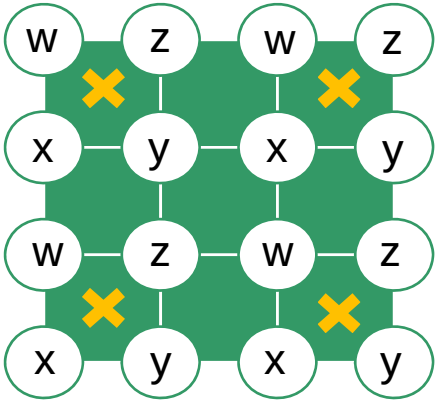
9 PCF samples plus some ALU right ?

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 3

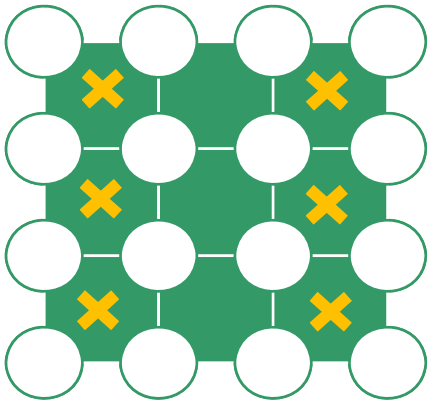
Let's filter a 4x4 visibility sample block using unique weights

Direct3D 10.1



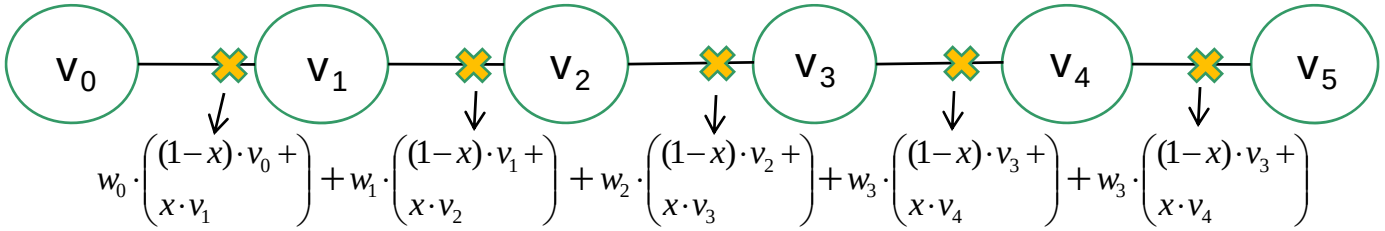
4 Gather() samples plus some ALU => (N/2)x(N/2) Gather() samples for NxN

Direct3D 10.0



6 shifted PCF samples plus post weight factors is enough => (N/2)x(N-1) PCF samples for NxN

Advanced Direct3D 10.1 shadow filtering 4

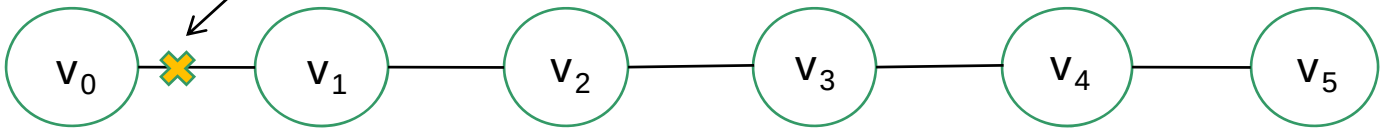


$$w_0 \cdot \begin{pmatrix} (1-x)\cdot v_0 + \\ x \cdot v_1 \end{pmatrix} + w_1 \cdot \begin{pmatrix} (1-x)\cdot v_1 + \\ x \cdot v_2 \end{pmatrix} + w_2 \cdot \begin{pmatrix} (1-x)\cdot v_2 + \\ x \cdot v_3 \end{pmatrix} + w_3 \cdot \begin{pmatrix} (1-x)\cdot v_3 + \\ x \cdot v_4 \end{pmatrix} + w_3 \cdot \begin{pmatrix} (1-x)\cdot v_3 + \\ x \cdot v_4 \end{pmatrix}$$

$$(1 - x) \cdot v_0 \cdot w_0 + \left(\sum_{k=1}^{N-2} v_k \cdot \left((w_{k-1} - w_k) \cdot x + w_k \right) \right) + x \cdot v_{N-1} \cdot w_{N-2}$$

left

$$\begin{pmatrix} (1-x)\cdot v_0 \cdot w_0 + \\ v1 \cdot (w1 + x \cdot (w_0 - w_1)) \end{pmatrix}$$



$$(1 - x') \cdot wp \cdot v_0 = (1-x) \cdot w_0 \cdot v_0$$

$$x' \cdot wp \cdot v_1 = v_1 \cdot (x \cdot (w_0 - w_1) + w_1)$$

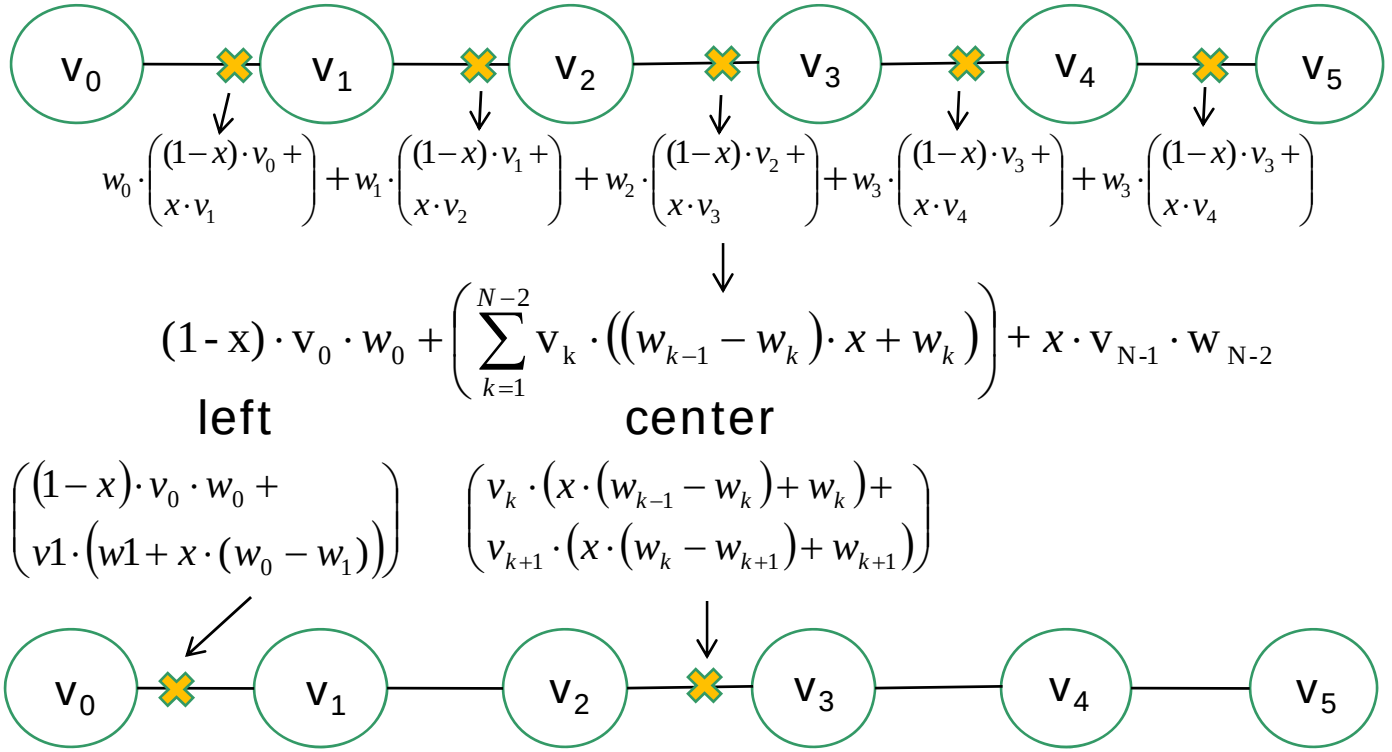
$$\longrightarrow$$

$$x' = \frac{(w_1 - w_0) \cdot x - w_1}{w_1 \cdot x - w_1 - w_0}$$

$$wp = -w_1 \cdot x + w_1 + w_0$$

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 4



$$w_0 \cdot \begin{pmatrix} (1-x)\cdot v_0 + \\ x \cdot v_1 \end{pmatrix} + w_1 \cdot \begin{pmatrix} (1-x)\cdot v_1 + \\ x \cdot v_2 \end{pmatrix} + w_2 \cdot \begin{pmatrix} (1-x)\cdot v_2 + \\ x \cdot v_3 \end{pmatrix} + w_3 \cdot \begin{pmatrix} (1-x)\cdot v_3 + \\ x \cdot v_4 \end{pmatrix} + w_3 \cdot \begin{pmatrix} (1-x)\cdot v_3 + \\ x \cdot v_4 \end{pmatrix}$$

$$(1 - x) \cdot v_0 \cdot w_0 + \left(\sum_{k=1}^{N-2} v_k \cdot \left((w_{k-1} - w_k) \cdot x + w_k \right) \right) + x \cdot v_{N-1} \cdot w_{N-2}$$

left

center

$$\begin{pmatrix} (1-x)\cdot v_0 \cdot w_0 + \\ v1 \cdot (w1 + x \cdot (w_0 - w_1)) \end{pmatrix} \qquad \begin{pmatrix} v_k \cdot (x \cdot (w_{k-1} - w_k) + w_k) + \\ v_{k+1} \cdot (x \cdot (w_k - w_{k+1}) + w_{k+1}) \end{pmatrix}$$

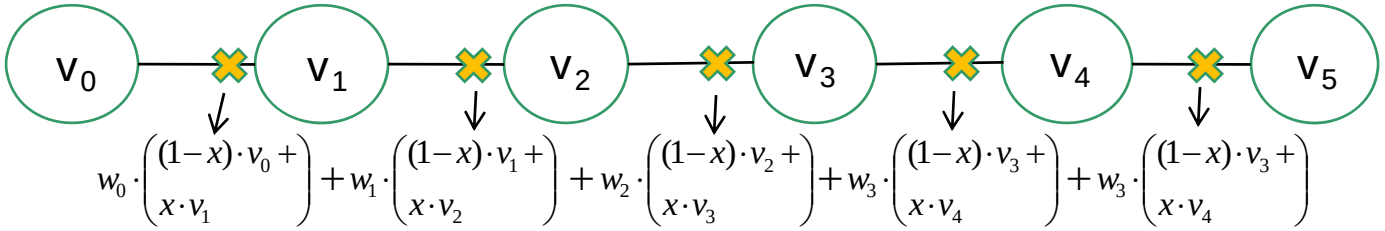
$$(1 - x') \cdot wp \cdot v_k = v_k \cdot (x \cdot (w_{k-1} - w_k) + w_k)$$
$$x' \cdot wp \cdot v_{k+1} = v_{k+1} \cdot (x \cdot (w_k - w_{k+1}) + w_{k+1})$$

$$\longrightarrow \quad x' = \frac{(w_{k+1} - w_k) \cdot x - w_{k+1}}{(w_{k+1} - w_{k-1}) \cdot x - w_{k+1} - w_k}$$

$$wp = (w_{k-1} - w_{k+1}) \cdot x + w_{k+1} + w_k$$

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 4

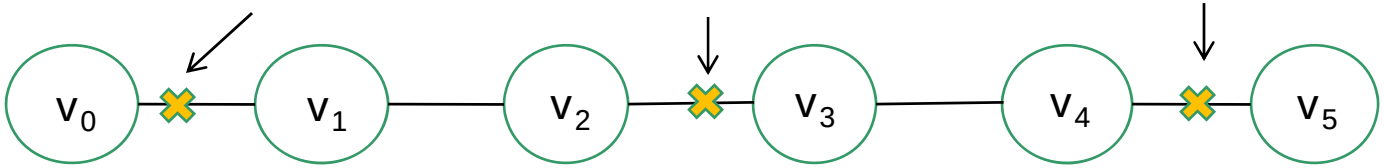


$$w_0 \cdot \begin{pmatrix} (1-x)\cdot v_0 + \\ x \cdot v_1 \end{pmatrix} + w_1 \cdot \begin{pmatrix} (1-x)\cdot v_1 + \\ x \cdot v_2 \end{pmatrix} + w_2 \cdot \begin{pmatrix} (1-x)\cdot v_2 + \\ x \cdot v_3 \end{pmatrix} + w_3 \cdot \begin{pmatrix} (1-x)\cdot v_3 + \\ x \cdot v_4 \end{pmatrix} + w_3 \cdot \begin{pmatrix} (1-x)\cdot v_3 + \\ x \cdot v_4 \end{pmatrix}$$

$$(1 - x) \cdot v_0 \cdot w_0 + \left(\sum_{k=1}^{N-2} v_k \cdot \left((w_{k-1} - w_k) \cdot x + w_k \right) \right) + x \cdot v_{N-1} \cdot w_{N-2}$$

left center right

$$\begin{pmatrix} (1-x)\cdot v_0 \cdot w_0 + \\ v1 \cdot (w1 + x \cdot (w_0 - w_1)) \end{pmatrix} \quad \begin{pmatrix} v_k \cdot (x \cdot (w_{k-1} - w_k) + w_k) + \\ v_{k+1} \cdot (x \cdot (w_k - w_{k+1}) + w_{k+1}) \end{pmatrix} \quad \begin{pmatrix} v_{N-2} \cdot (x \cdot (w_{N-2} - w_{N-1}) + w_{N-1}) + \\ x \cdot w_{N-1} v_{N-1} \end{pmatrix}$$



$$(1 - x') \cdot wp \cdot v_{N-2} = v_{N-2} \cdot (x \cdot (w_{N-2} - w_{N-1}) + w_{N-1})$$

$$x' \cdot wp \cdot v_{N-1} = x \cdot w_{N-1} \cdot v_{N-1}$$

$$x' = \frac{w_{N-1} \cdot x}{w_{N-2} \cdot x + w_{N-1}}$$

$$wp = w_{N-2} \cdot x + w_{N-1}$$

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Advanced Direct3D 10.1 shadow filtering 5

- Direct3D 10.0
  - needs (N/2)x(N-1) PCF samples for Gaussian shadows – not (N-1)x(N-1)!
  - can do one row with (N/2) samples with shifted x texture coords
  - y texture coord stays untouched
- Stats of an optimized shader for 8x8
  - Direct3D 10.1 shader roughly twice as fast as the Direct3D10.0 version
  - Direct3D 10.1 shader as fast as the optimized uniform (N/2)x(N/2) filter under Direct3D10.0


From DICE’s Frostbite Engine:
Standard 2x2 shadow filtering


From DICE’s Frostbite Engine:
5x5 Gaussian filtering

Tom Clancy‘s HAWX
Publisher: Ubisoft
Developer: Ubisoft Romania
Normal Quality – Blurred VSM


Tom Clancy‘s HAWX
Publisher: Ubisoft
Developer: Ubisoft Romania
Gaussian Shadows
ATK 1 2 DEF
Reach downtown
1196 KMH
ALT 1444
Rio
100%
35929m
JStrike 240
100

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Stormrise, Publisher: SEGA
Developer: The Creative Assembly Australia
Normal Shadow Quality

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Stormrise, Publisher: SEGA
Developer: The Creative Assembly Australia
Gaussian Shadows

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Summary: 1

- HDAO adds enourmous depth to the scene, at an affordable cost
- Using Direct3D 10.1 `gather4` instruction greatly accelerates performance
- Growing number of game developers using the effect
- Mail jon.story@amd.com if you would like to know more...

THINK SERVICES
GDC 09
learn network inspire
www.GDConf.com

Summary: 2

- Conventional high quality shadow filtering is suprisingly fast
  - Even under Direct3D 10.0/9
- Direct3D 10.1 delivers the best performance
  - No reason not to use gaussian shadows!
  - Direct3D 11 supports Gather()!
- Mail holger.gruen@amd.com if you want the shaders or the derivations for (N/2)x(N/2) PCF sample shadows

Questions?

Please fill in the feedback forms...