AMD HIGH DEFINITION AMBIENT OCCLUSION (HDAO) LIBRARY

ALEX KHARLAMOV

TABLE OF CONTENTS

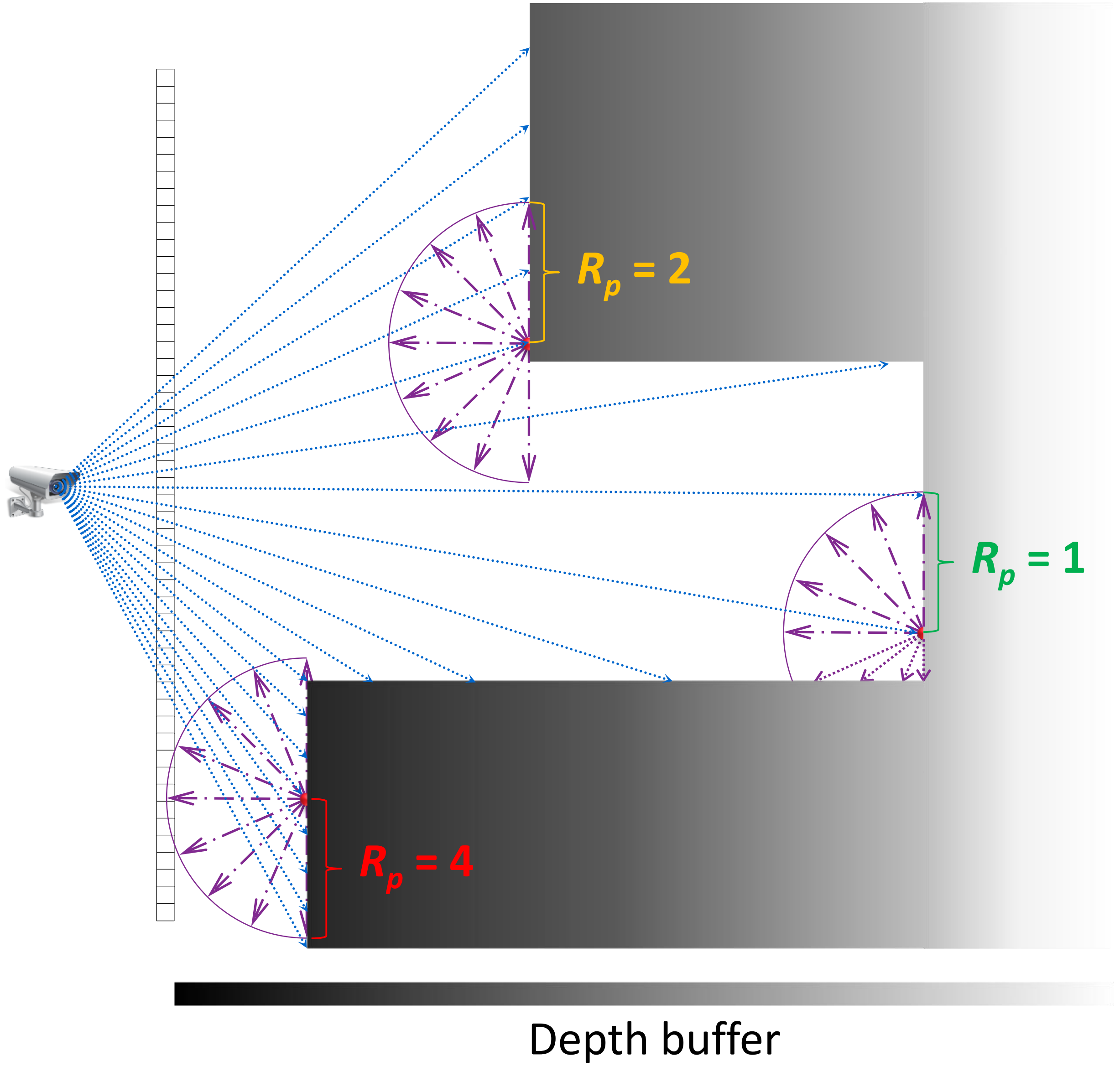
HDAO TECHNIQUE

TRIVIA AND HDAO V1.0 ALGORITHM DESCRIPTION

TRIVIA

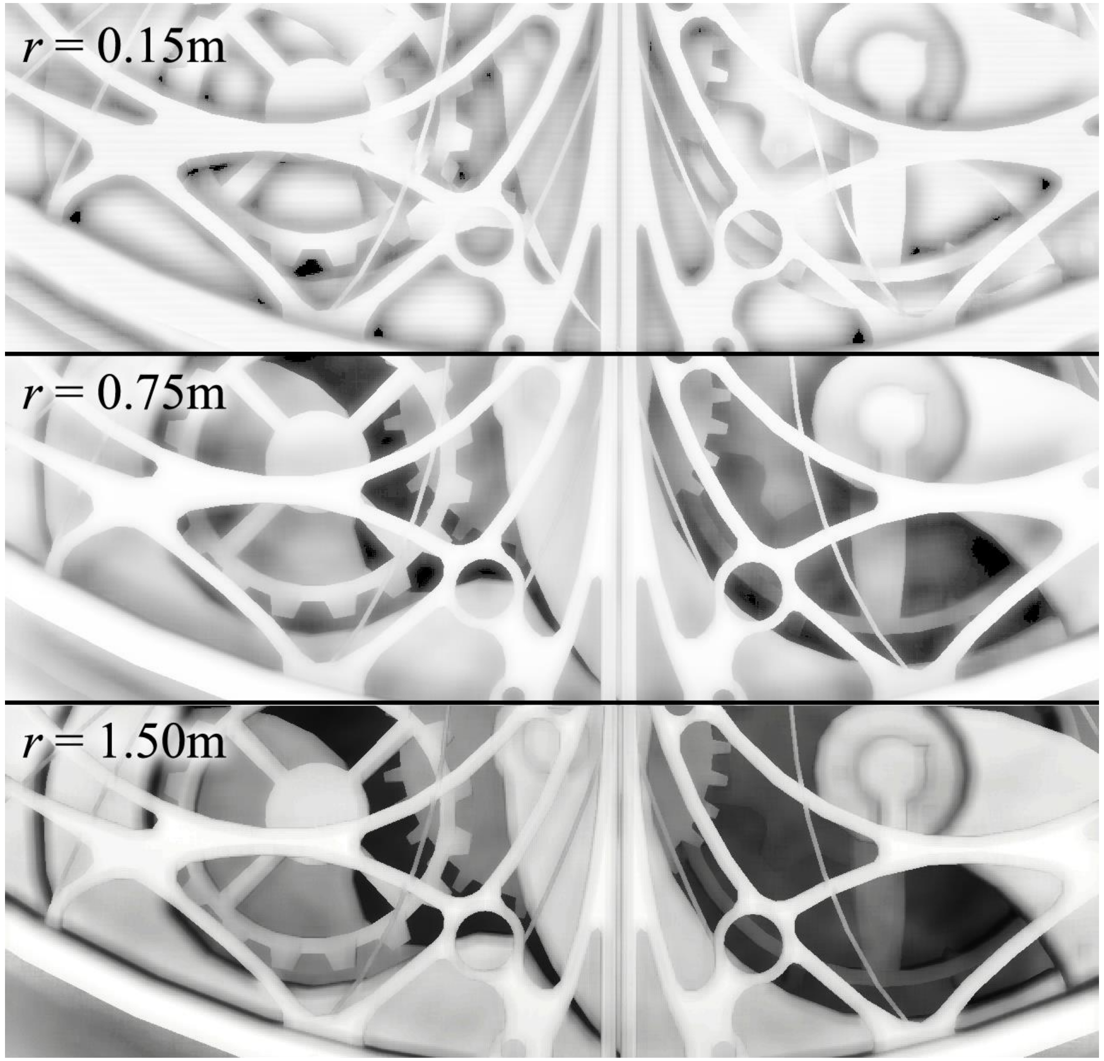
AMD

- AO evaluates how exposed is a given point to ambient light
- AO at point ***P*** with normal ***n*** is typically computed as an integral of visibility function over the hemisphere
- In screen-space AO techniques, fixed world space radius ***R*** results in variable screen space radius R_p creating uneven texture access pattern
  - Sampling becomes worse with jittered sampling
- In practice adjusting hemisphere radius ***R*** produces different visual results
  - Mostly artistic parameter



TRIVIA

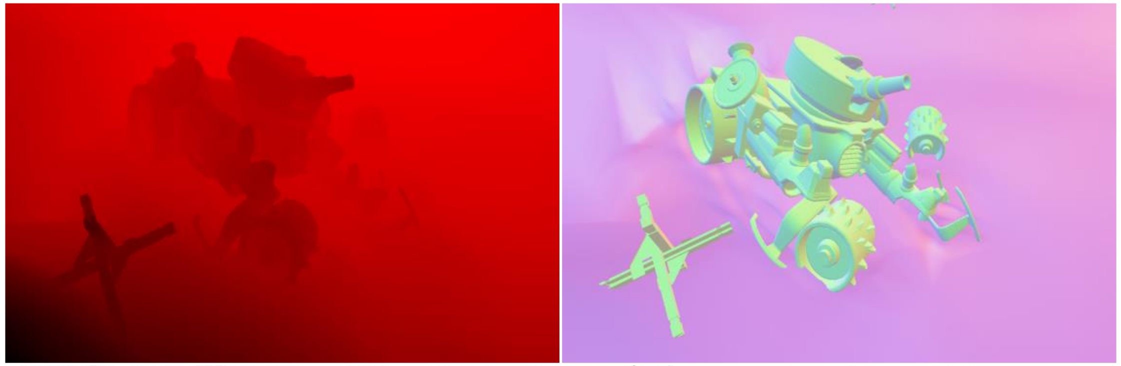
- AO evaluates how exposed is a given point to ambient light
- AO at point ***P*** with normal ***n*** is typically computed as an integral of visibility function over the hemisphere
- In screen-space AO techniques, fixed world space radius ***R*** results in variable screen space radius R_p creating uneven texture access pattern
  - Sampling becomes worse with jittered sampling
- In practice adjusting hemisphere radius ***R*** produces different visual results
  - Mostly artistic parameter

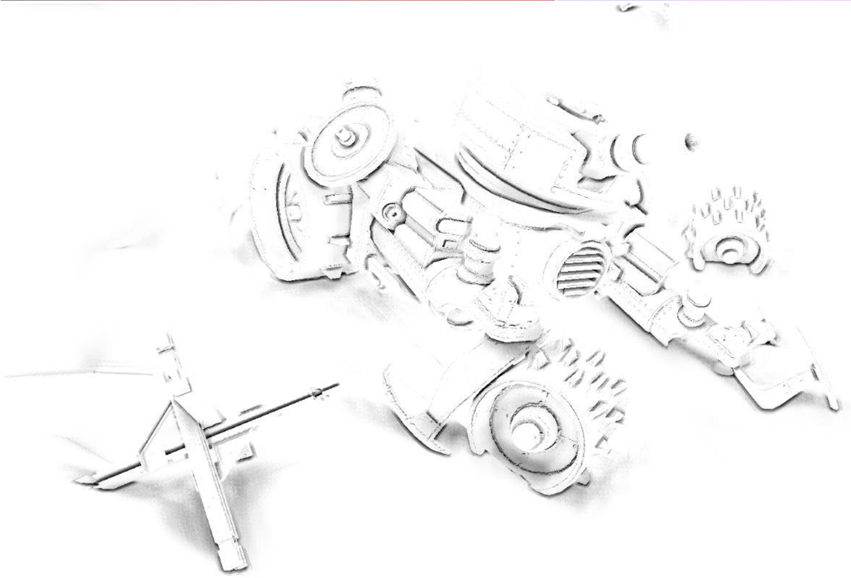


HDAO TECHNIQUE

- Screen space AO
  - Can use either Depth or Depth + Normal as input
  - Computes view space position
    - If Normal is available, position is displaced along the normal
- Compute shader implementation
  - Uses group shared memory to cache a block of texels
  - Targets DX12 asynchronous compute

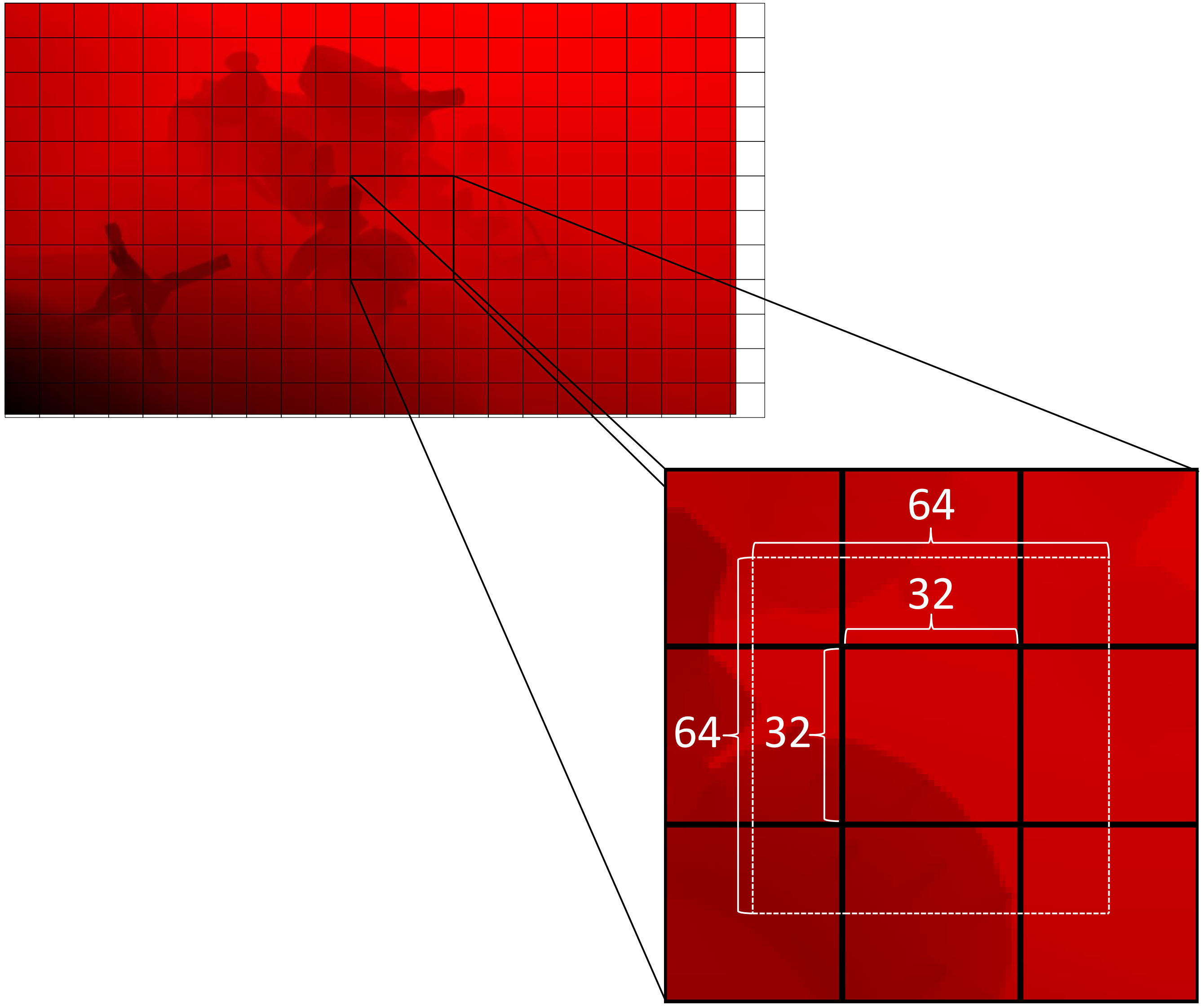
Input





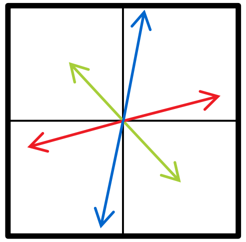
HDAO TECHNIQUE

- Tile input depth (and normal) texture(s) into tiles of 32x32 texels
- For each tile launch a compute thread group of 32x32 threads, each thread corresponding to one texel
- For each thread group load corresponding tile of texels into shared memory, with a 16-texel apron
  - 64^2 texels loaded
  - Actually storing view space position, compressed into uint2 with xy using fp16 precision
- 16-texel apron implies a limit on sampling range, i.e. making AO radius variable *in world space*

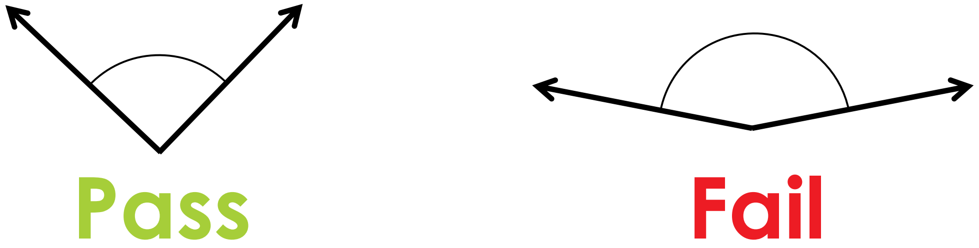


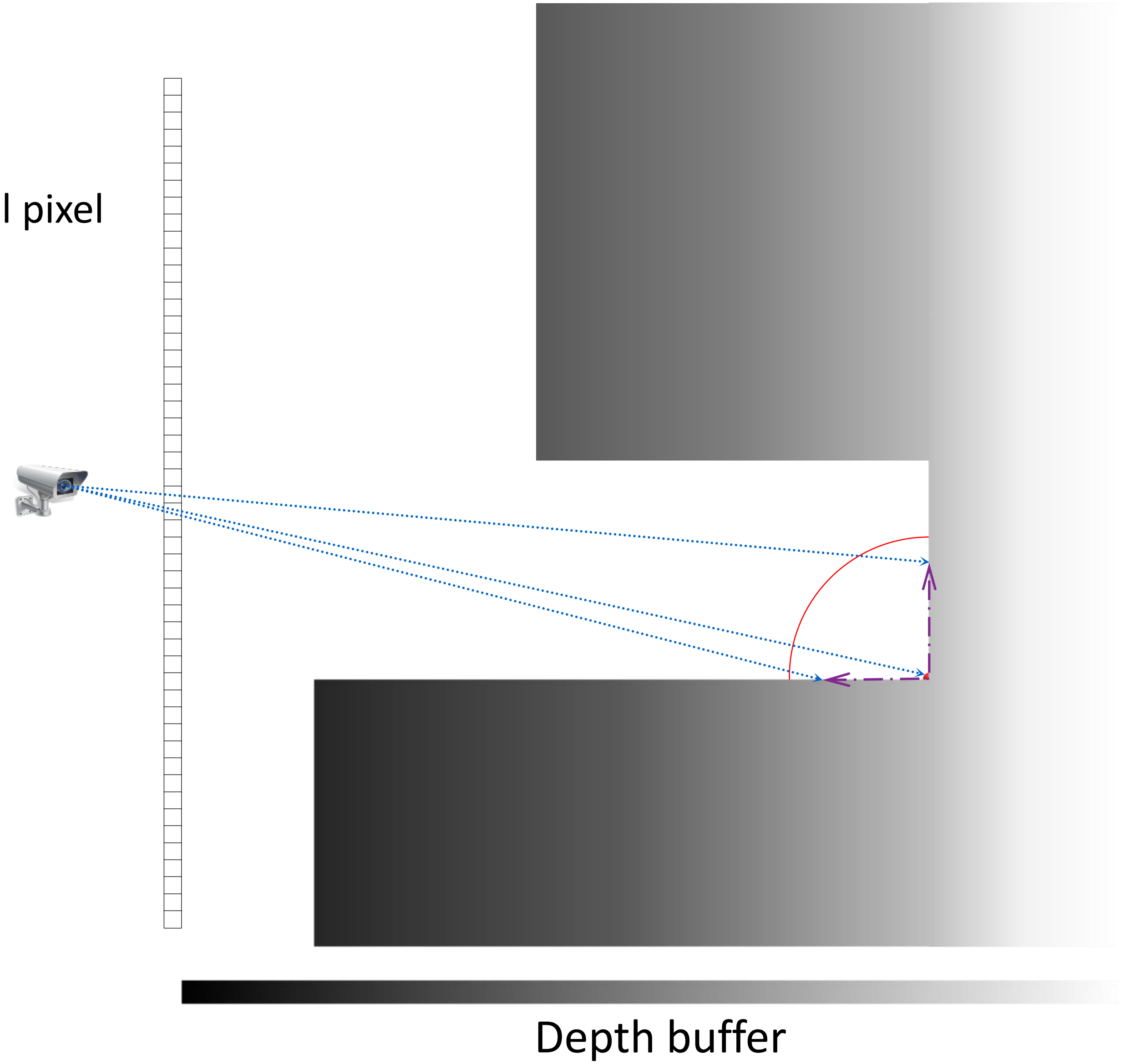
HDAO KERNEL

- Detect valley in camera space
  - Sample the central pixel of interest
  - Sample a pair of pixels, mirrored through the central pixel



  - If both values of a pair fall within a specific radius
  - And are closer to the camera than the central pixel
  - Then we have detected a valley
    - Reject if angle is too shallow





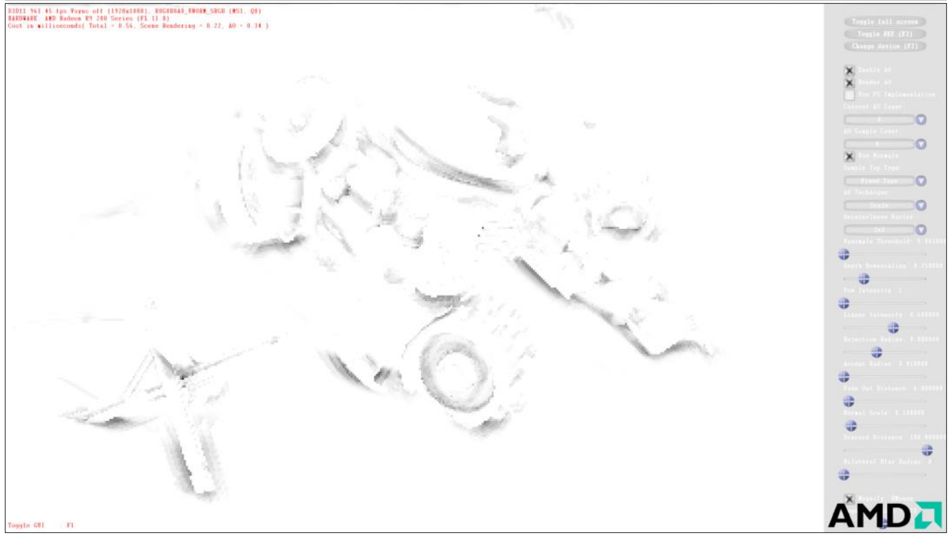
HDAO TECHNIQUE (V 1.0)

HDAO VISUALS = FUNC(INPUT DEPTH BUFFER SCALING)

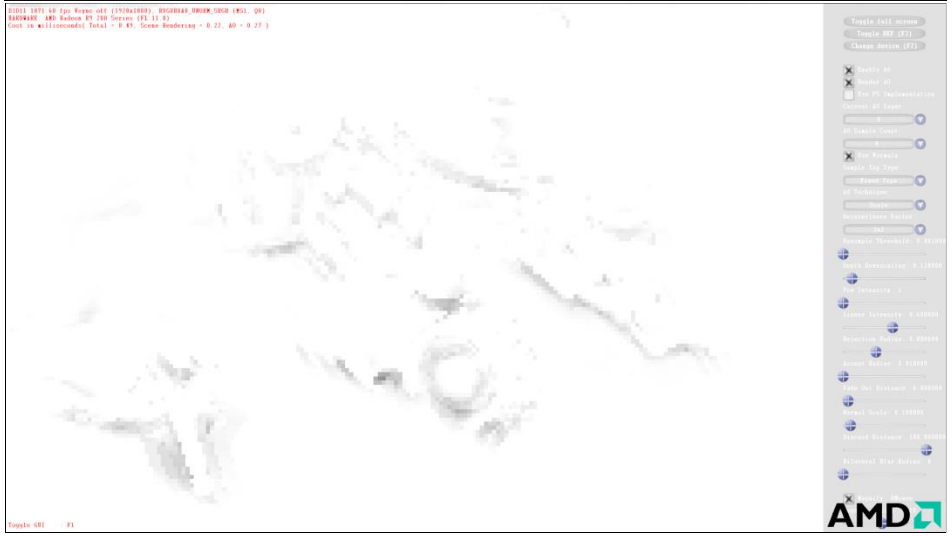
No scaling

0.5 scaling

0.25 scaling



0.12 scaling



HDAO TECHNIQUE (V 1.0)

MULTI-RES APPROACH, COMBINE 3 LAYERS = {NO SCALING, 0.25 SCALING, 0.12 SCALING} | 1.18 MS @ 1080P

HDAO TECHNIQUE (V 1.0)

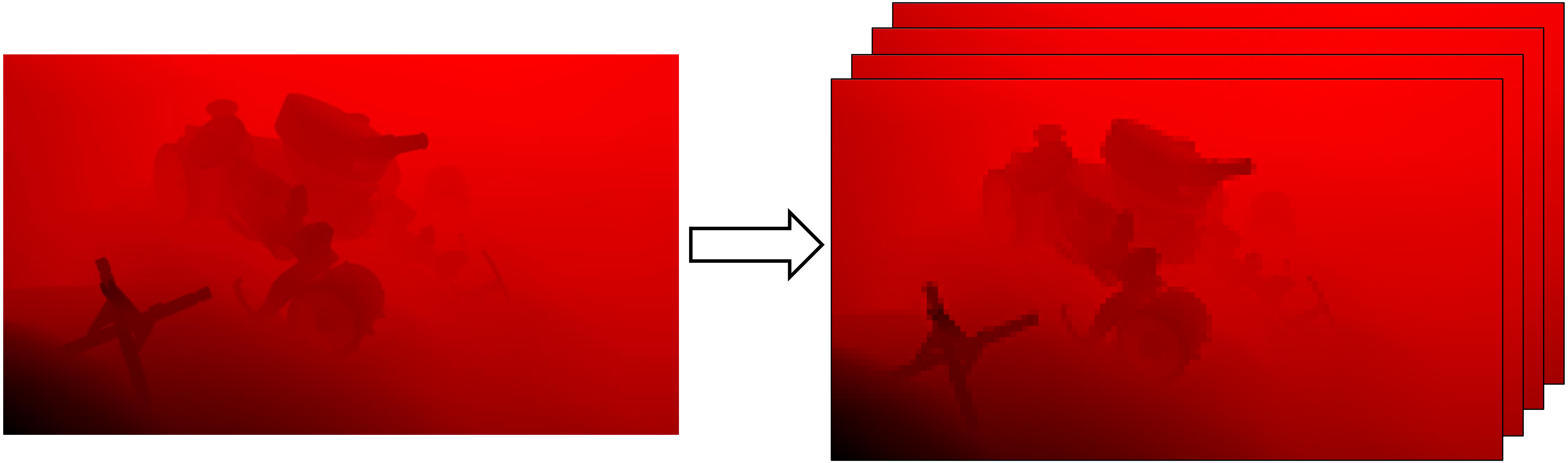
- Multi-res approach
  - PROS: captures details at various resolution levels
  - CONS: artifacts related to downscaling: aliasing and flickering
- Bilateral blurring helps reduce aliasing
  - But flickering is still present

HDAO TECHNIQUE

WORK AND INPUT DE-INTERLEAVING

HDAO TECHNIQUE (V 2.0)

- Use input data de-interleaving instead of downscaling
  - As described by Louis Bavoil (http://www.gdcvault.com/play/1017623/Advanced-Visual-Effects-with-DirectX)
  - Given a de-interleaving pattern size of NxN de-interleave original input image into NxN smaller copies



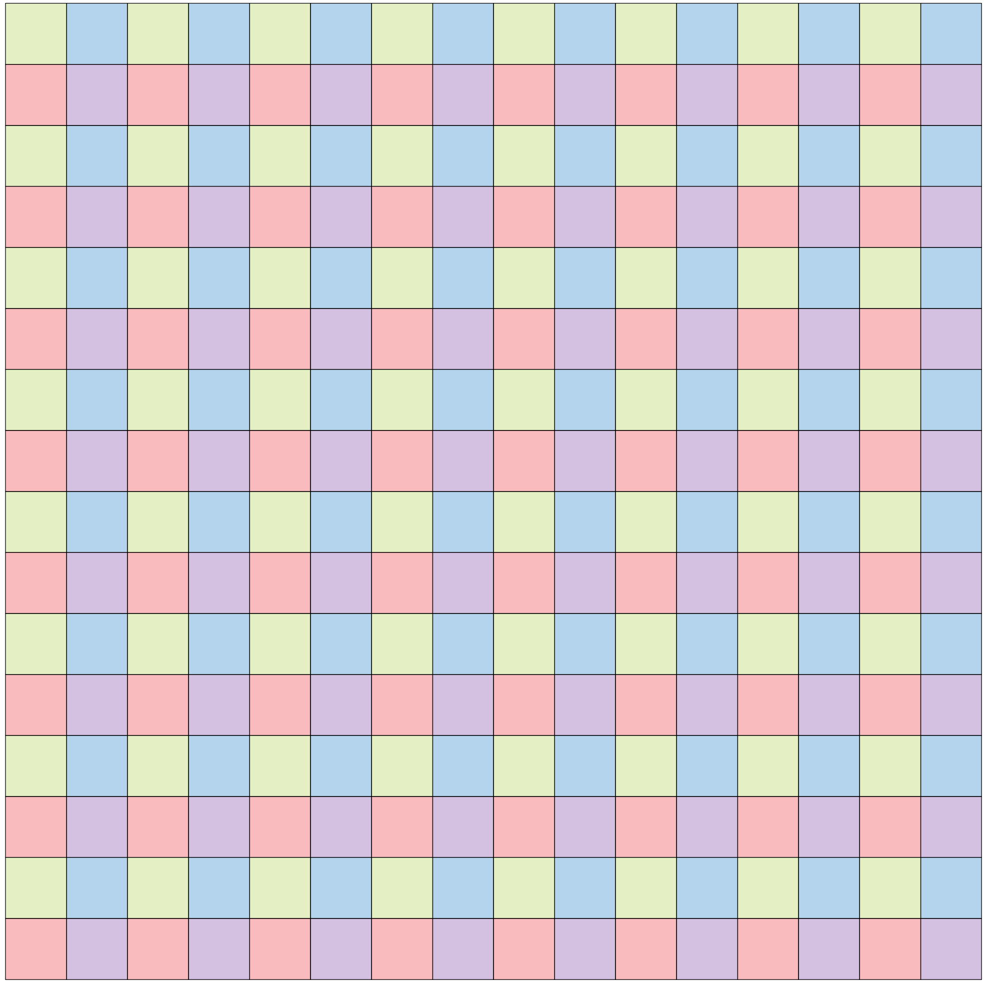
DE-INTERLEAVING TEXTURE INPUT

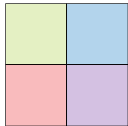
AN ARRAY OF SMALLER RESOLUTION INPUTS

DE-INTERLEAVING TEXTURE INPUT

DE-INTERLEAVING TEXTURE INPUT

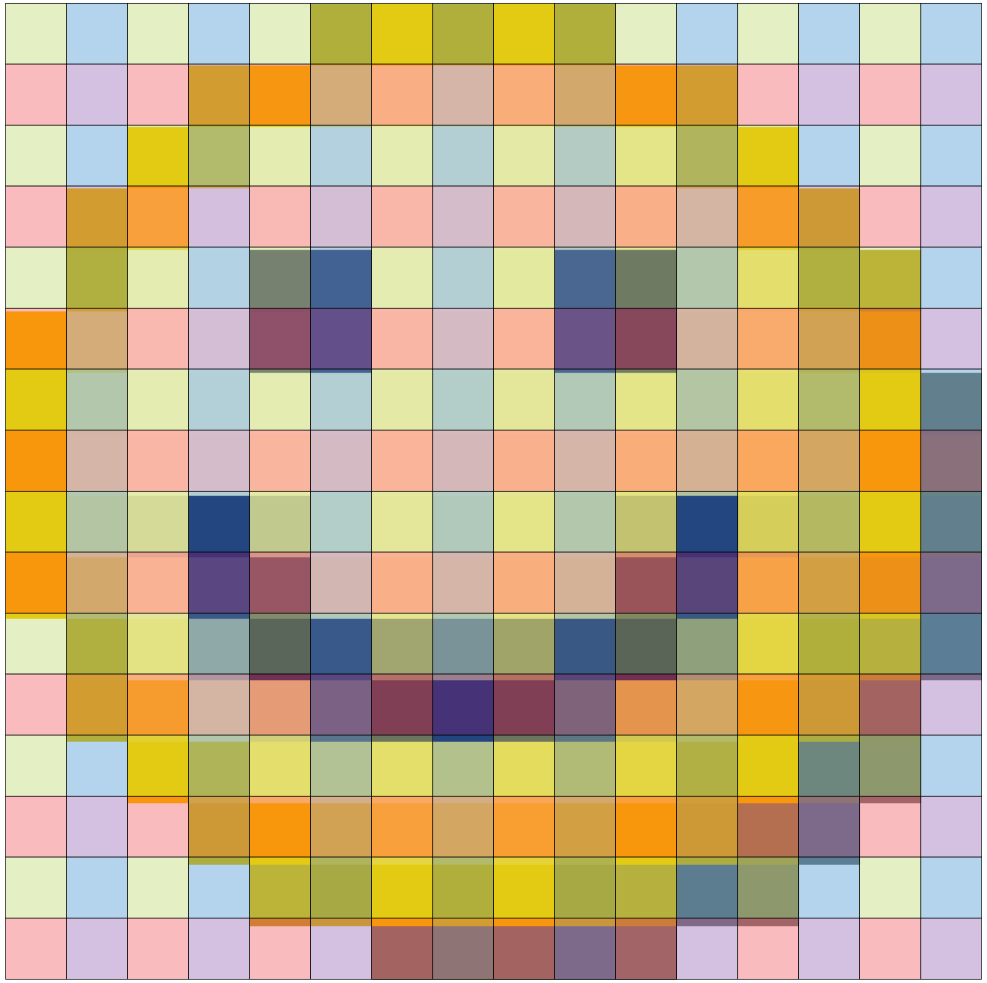
TILE THE IMAGE WITH A SELECTED DE-INTERLEAVING PATTERN (2X2 PATTERN BELOW)

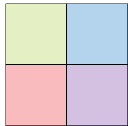




DE-INTERLEAVING TEXTURE INPUT

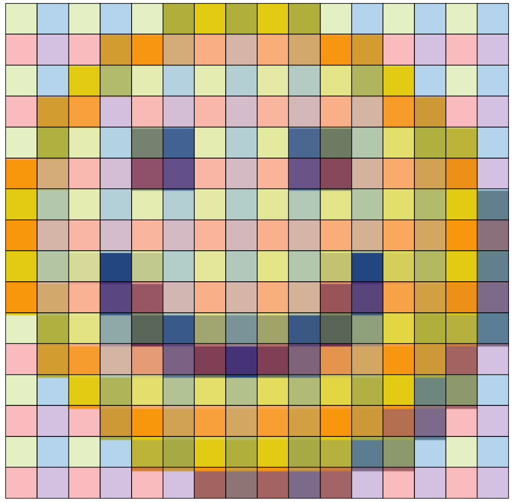
TILE THE IMAGE WITH A SELECTED DE-INTERLEAVING PATTERN (2X2 PATTERN BELOW)





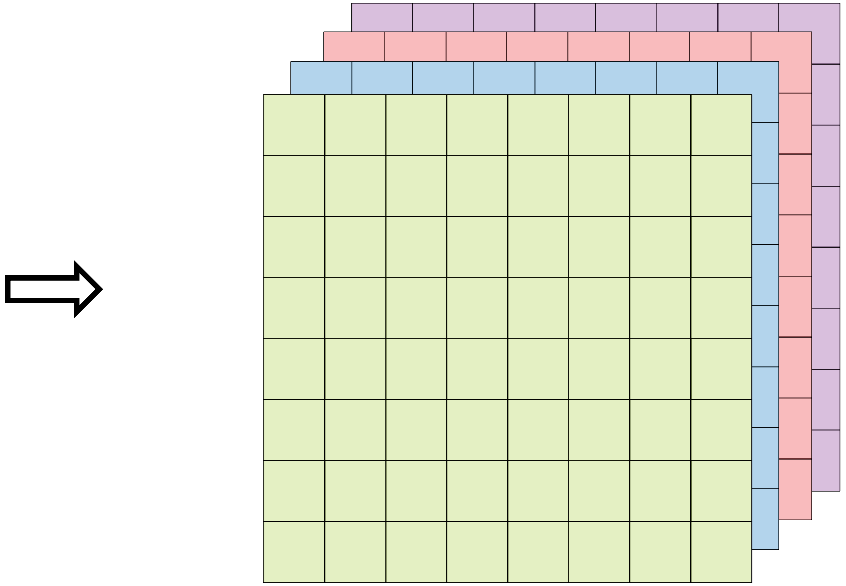
DE-INTERLEAVING TEXTURE INPUT

SEPARATE COLOR CODED TEXELS INTO SEPARATE TEXTURE RESOURCES



Source image

Width x Height resolution



De-interleaved texture array

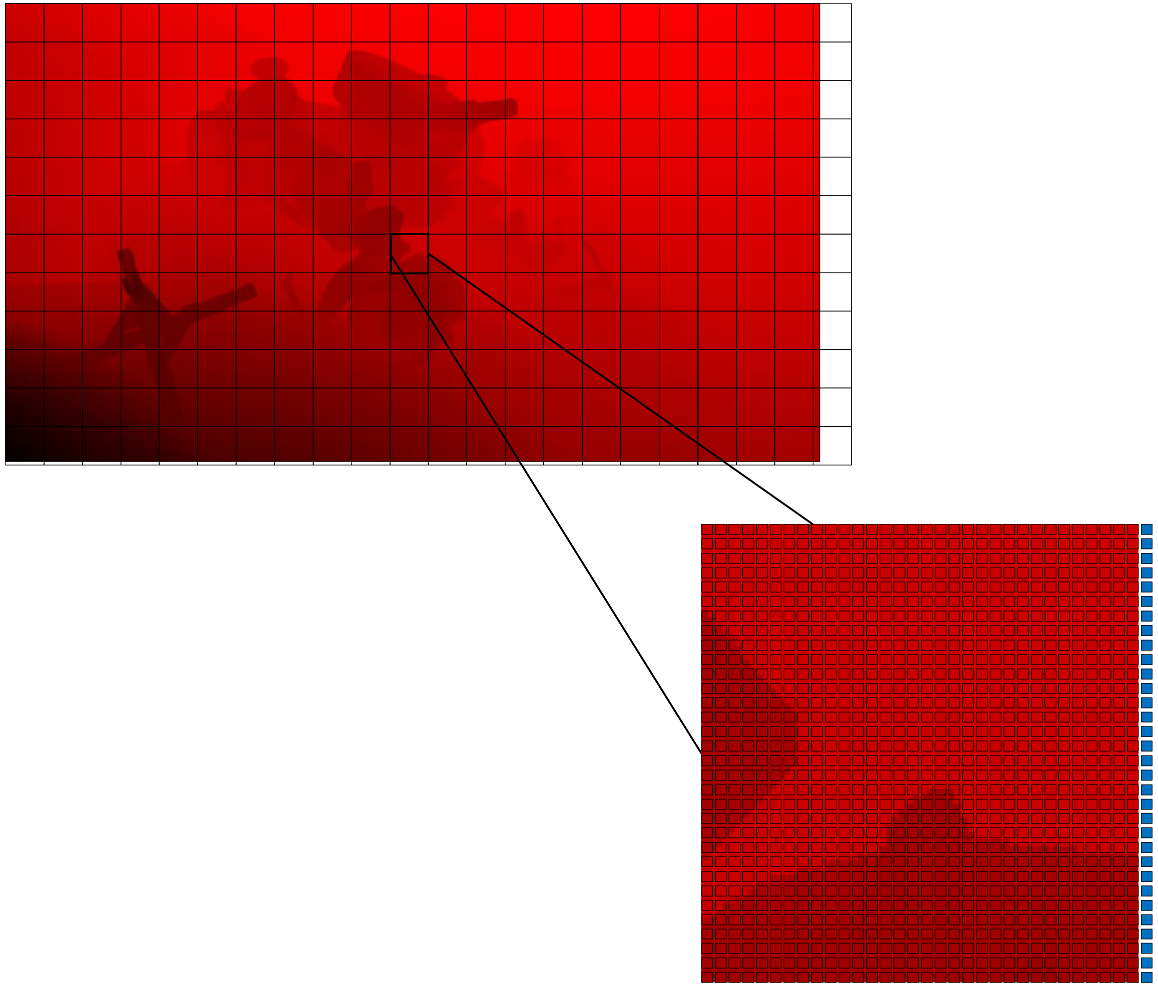
(De-interleaved Factor (DF))^2 array slices

(Width / DF) x (Height / DF) resolution each

DE-INTERLEAVING TEXTURE INPUT

COMPUTE SHADER IMPLEMENTATION (1/2)

- Tile input resource into blocks of 32x32 texels
- Launch a thread group per tile, with 32x32 threads
- Load 32x32 texels into shared memory
  - Depth is converted to view space z
  - In case of depth+normal input it is converted to view space position with displacement
  - Shared memory array is actually 32x33 to avoid bank conflicts on subsequent reads
- Group Sync
- Rearrange threads to write de-interleaved input into a texture array
  - R16 (or RGBA16) format

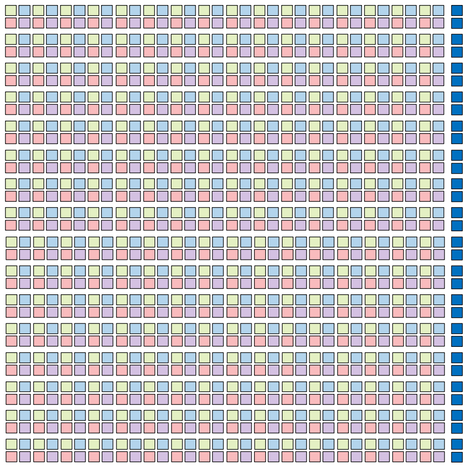


Shared memory array

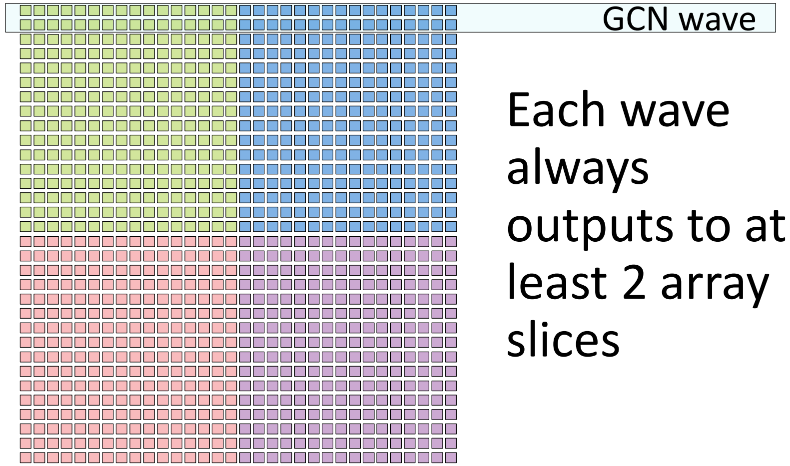
DE-INTERLEAVING TEXTURE INPUT

COMPUTE SHADER IMPLEMENTATION (2/2)

- Rearrange threads to write de-interleaved input into a texture array
  - Attempt to match some sort of a rectangular shaped output into a texture 2D array slices
  - Multiple ways to do this
    - Different ways to rearrange threads may change the way de-interleaved data get’s written to a texture array
    - In practice it doesn’t seem to affect performance (on GCN 2 & 3)
  - The extra column in shared memory helps avoid bank conflicts when reading

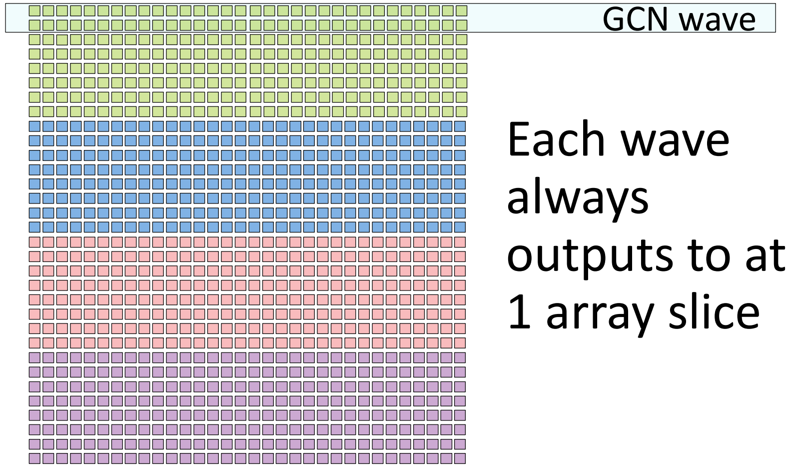


Shared memory array



Each wave always outputs to at least 2 array slices

OR



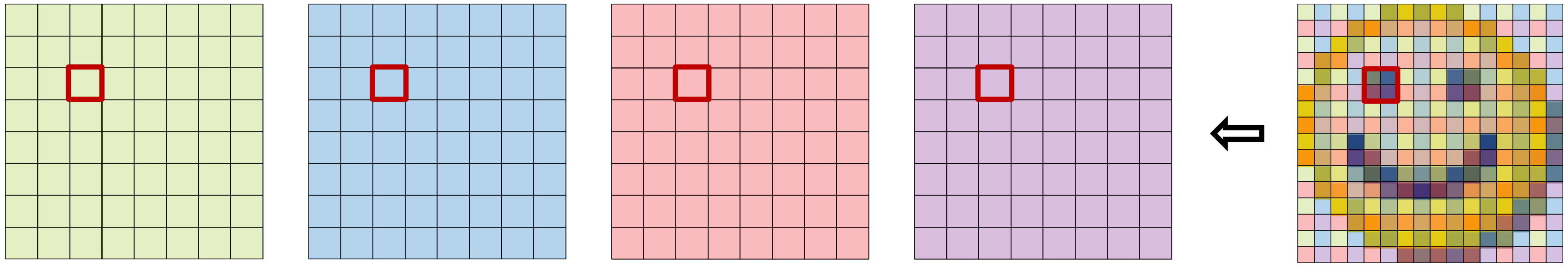
Each wave always outputs to 1 array slice

Rearranged threads

DE-INTERLEAVING AO COMPUTATION

RUN THE HDAO SHADER USING DE-INTERLEAVED TEXTURE2D ARRAY AS INPUT

- Launch all thread groups for all slices together, calculate array slice index based on group index
- Take care to compute any re-projections correctly
  - Texels at the same position (u, v) from different slices had different positions in the original input
  - Texel offsets can be computed from slice index



- Produces multiple smaller AOs
  - Shader writes directly into a full-resolution Texture2D
  - Writes are scattered but don’t seem to affect performance (GCN 3)

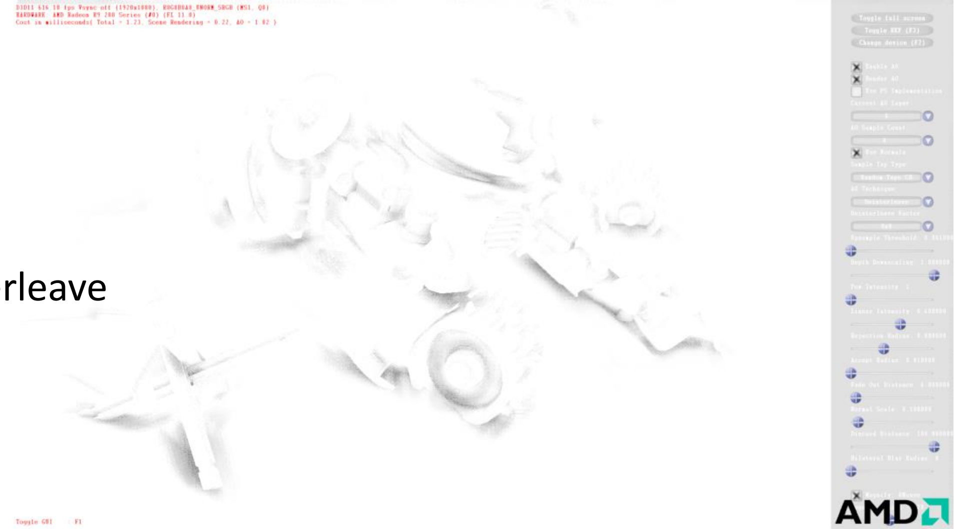
AO TECHNIQUE COMPARISON

No
scaling

2x2
De-interleave

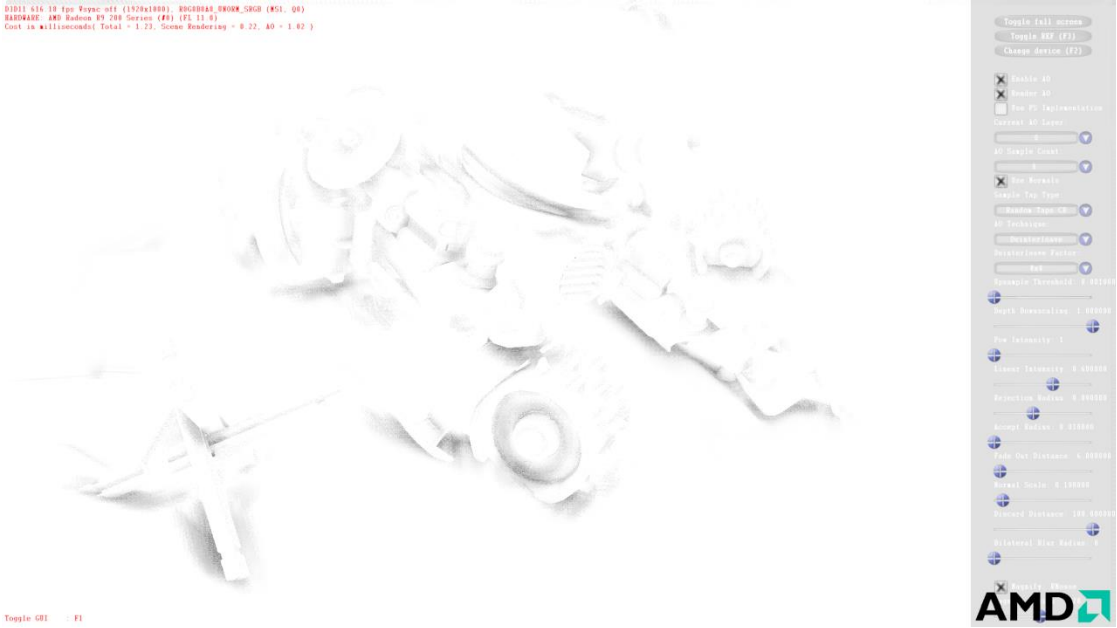
4x4
De-interleave

8x8
De-interleave

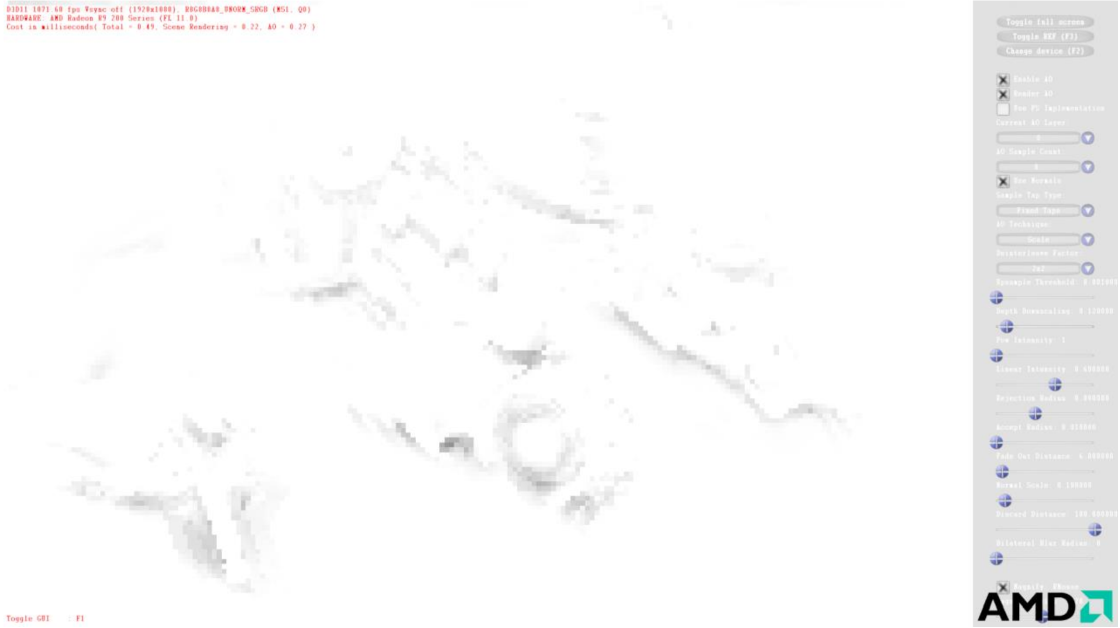


AO TECHNIQUE COMPARISON

8x8 de-interleave pattern



0.12 scaling



HDAO TECHNIQUE (V 2.0)

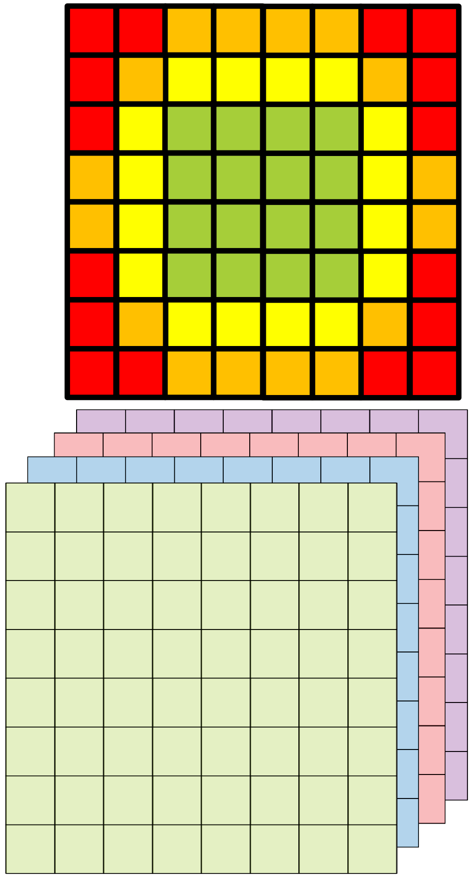
MULTI-RES APPROACH, COMBINE 2 LAYERS = {NO SCALING, 8X8 DE-INTERLEAVED} | 1.55MS @1080P

HDAO TECHNIQUE

PERFORMANCE CONSIDERATIONS

HDAO PERFORMANCE

- A wide variety of performance parameters
  - Input: {Depth | Depth Input}
  - Sampling pattern mode: {Fixed | Random (read from constant buffer) | Random (read from texture)}
  - Sample count: {8, 16, 24, 32}
    - Higher sample count covers larger area
    - De-interleaving factors: {2|4|8} or down scaling
  - Early out: {Fade out distance| Discard distance}
- Multiple shader permutation
- Match de-interleave pattern to random sampling pattern
  - Random sampling becomes uniform within each slice of work



HDAO PERFORMANCE

MIX AND MATCH TECHNIQUES FOR BEST PERFORMANCE & VISUAL BALANCE

Original HDAO, no downscaling						
Resolution	Sampling Pattern					
1920 x 1080	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	0.76	0.85	0.88	0.98	0.86	0.96
16 taps	1.18	1.3	1.33	1.44	1.32	1.42
24 taps	1.59	1.69	1.78	1.9	1.77	1.87
32 taps	1.99	2.1	2.24	2.35	2.21	2.32
2560 x 1600	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	1.41	1.63	1.65	1.87	1.62	1.84
16 taps	2.23	2.46	2.53	2.75	2.51	2.72
24 taps	3.03	3.24	3.41	3.64	3.39	3.6
32 taps	3.84	4.05	4.3	4.52	4.26	4.47

De-interleaved HDAO, 2x2 pattern						
Resolution	Sampling Pattern					
1920 x 1080	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	0.75	0.88	0.83	0.96	0.87	1
16 taps	1.16	1.28	1.29	1.42	1.35	1.47
24 taps	1.6	1.73	1.75	1.88	1.79	1.92
32 taps	1.99	2.12	2.21	2.35	2.26	2.38
2560 x 1600	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	1.39	1.64	1.55	1.8	1.63	1.88
16 taps	2.19	2.43	2.44	2.69	2.56	2.8
24 taps	3.07	3.31	3.33	3.58	3.42	3.66
32 taps	3.83	4.06	4.23	4.48	4.33	4.56

HDAO PERFORMANCE

AMD

	De-interleaved HDAO, 4x4 pattern					
Resolution	Sampling Pattern					
1920 x 1080	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	0.77	0.89	0.85	0.99	0.89	1.02
16 taps	1.17	1.3	1.31	1.44	1.36	1.48
24 taps	1.63	1.76	1.77	1.9	1.8	1.93
32 taps	2.02	2.14	2.23	2.36	2.26	2.38
2560 x 1600	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	1.42	1.66	1.58	1.85	1.67	1.89
16 taps	2.22	2.45	2.47	2.72	2.59	2.81
24 taps	3.1	3.34	3.37	3.61	3.45	3.68
32 taps	3.86	4.09	4.27	4.52	4.34	4.57

	De-interleaved HDAO, 8x8 pattern					
Resolution	Sampling Pattern					
1920 x 1080	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	0.9	1.03	1	1.13	1.03	1.16
16 taps	1.33	1.45	1.5	1.63	1.54	1.66
24 taps	1.81	1.94	1.99	2.13	2.01	2.13
32 taps	2.23	2.36	2.49	2.63	2.5	2.62
2560 x 1600	Fixed		Random (CB)		Random (SRV)	
	Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
8 taps	1.53	1.85	1.69	2.04	1.77	2.06
16 taps	2.33	2.63	2.58	2.92	2.68	2.98
24 taps	3.19	3.52	3.48	3.81	3.54	3.85
32 taps	3.95	4.25	4.39	4.71	4.44	4.74

HDAO PERFORMANCE

Resolution 1920 x 1080		Deinterleaved 2x2 / Scaled			
Fixed		Random (CB)		Random (SRV)	
Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
0.986842	1.035294	0.943182	0.979592	1.011628	1.041667
0.983051	0.984615	0.969925	0.986111	1.022727	1.035211
1.006289	1.023669	0.983146	0.989474	1.011299	1.026738
1	1.009524	0.986607	1	1.022624	1.025862

Resolution 2560 x 1600		Deinterleaved 2x2 / Scaled			
Fixed		Random (CB)		Random (SRV)	
Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
0.985816	1.006135	0.939394	0.962567	1.006173	1.021739
0.982063	0.987805	0.964427	0.978182	1.01992	1.029412
1.013201	1.021605	0.97654	0.983516	1.00885	1.016667
0.997396	1.002469	0.983721	0.99115	1.016432	1.020134

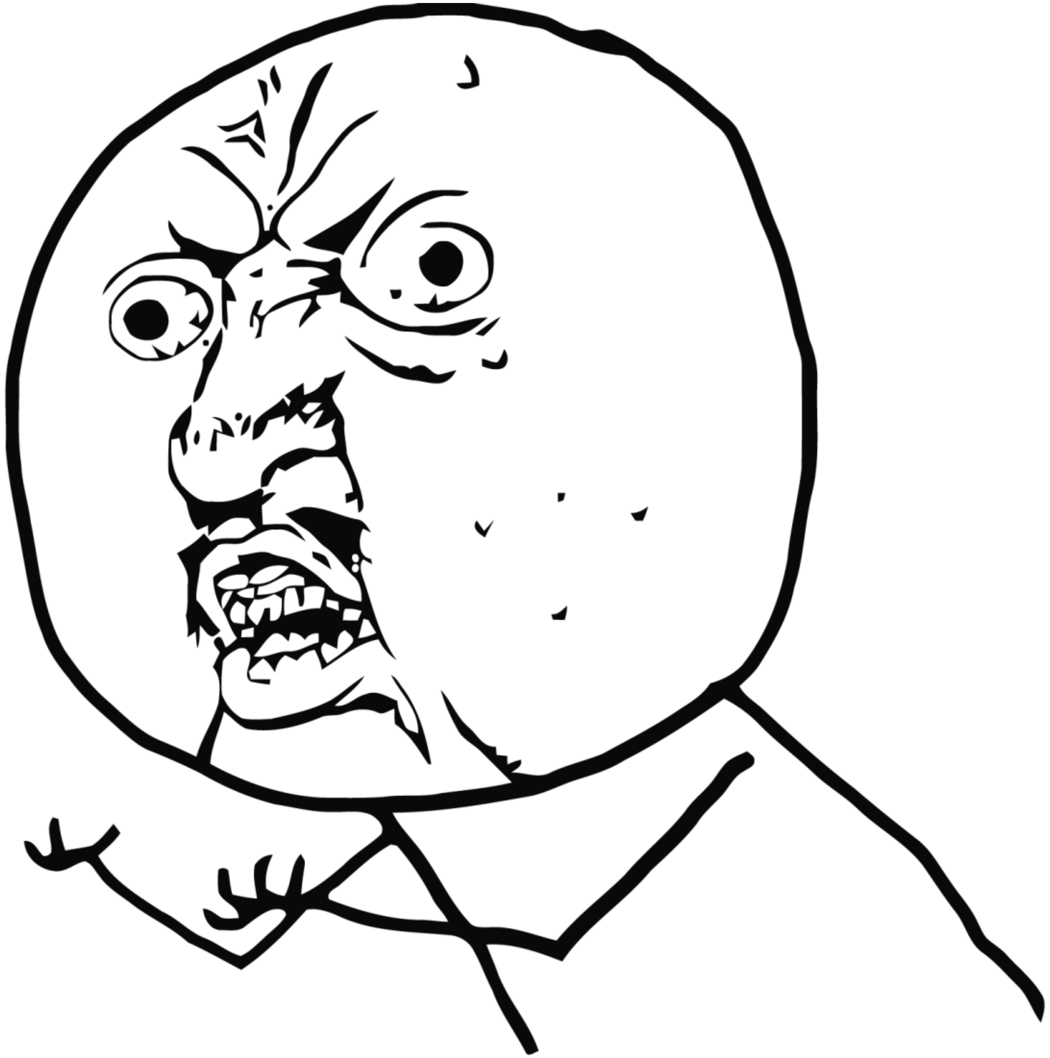
Resolution 1920 x 1080		Deinterleaved 4x4 / Scaled			
Fixed		Random (CB)		Random (SRV)	
Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
1.013158	1.047059	0.965909	1.010204	1.034884	1.0625
0.991525	1	0.984962	1	1.030303	1.042254
1.025157	1.04142	0.994382	1	1.016949	1.032086
1.015075	1.019048	0.995536	1.004255	1.022624	1.025862

Resolution 2560 x 1600		Deinterleaved 4x4 / Scaled			
Fixed		Random (CB)		Random (SRV)	
Depth	Depth Normal	Depth	Depth Normal	Depth	Depth Normal
1.007092	1.018405	0.957576	0.989305	1.030864	1.027174
0.995516	0.995935	0.976285	0.989091	1.031873	1.033088
1.023102	1.030864	0.98827	0.991758	1.017699	1.022222
1.005208	1.009877	0.993023	1	1.018779	1.022371

HDAO PERFORMANCE

DEINTERLEAVED 8X8 SEEMS TO BE SLOWER THAN 2X2 AND 8X8 PATTERNS

Resolution 1920 x 1080			Deinterleaved 8x8 / Scaled				
Fixed			Random (CB)			Random (SRV)	
Depth	Depth Normal		Depth	Depth Normal		Depth	Depth Normal
1.184211	1.211765		1.136364	1.153061		1.197674	1.208333
1.127119	1.115385		1.12782	1.131944		1.166667	1.169014
1.138365	1.147929		1.117978	1.121053		1.135593	1.139037
1.120603	1.12381		1.111607	1.119149		1.131222	1.12931
Resolution 2560 x 1600			Deinterleaved 8x8 / Scaled				
Fixed			Random (CB)			Random (SRV)	
Depth	Depth Normal		Depth	Depth Normal		Depth	Depth Normal
1.085106	1.134969		1.024242	1.090909		1.092593	1.119565
1.044843	1.069106		1.019763	1.061818		1.067729	1.095588
1.052805	1.08642		1.020528	1.046703		1.044248	1.069444
1.028646	1.049383		1.02093	1.042035		1.042254	1.060403



HDAO GPU PERF STUDIO COUNTER

DEI-NTERLEAVED 8X8 SEEMS TO BE SLOWER THAN 2X2 AND 4X4 PATTERNS

Deinterleaved 2x2

State Bucket	HDAO	De-interleave
GPUTime (ms)	0.507704	0.121185
GPUBusy (%)	100	100
CSBusy (%)	99.1175	52.6536
CSTime (ms)	0.503223	0.0638084
TexUnitBusy (%)	18.4891	25.3592
CSThreadGroups	2040	2040
CSVALUInsts	197.992	13.8991
CSVALUUtilization (%)	37.445	66.5639
CSVALUBusy (%)	34.4708	9.80697
CSSALUBusy (%)	3.19908	7.79348
CSMemUnitBusy (%)	18.4755	25.1065
CSMemUnitStalled (%)	0.377827	1.12837
CSFetchSize (kb)	4159.81	8100.75
CSWriteSize (kb)	2347	3795.81
CSCacheHit (%)	80.5065	25.0275
CSWriteUnitStalled (%)	0	0
CSGDSInsts	0	0
CSLDSInsts	9.07353	0.708701
CSALUStalledByLDS (%)	0.00100297	5.78E-05
CSLDSBankConflict (%)	0.845048	0

Deinterleaved 8x8

State Bucket	HDAO	De-interleave
GPUTime (ms)	0.6 | 0.6 (0%)	0.223 | 0.225 (0%)
GPUBusy (%)	100 | 100 (0%)	100 | 100 (0%)
CSBusy (%)	90.11 | 90.144 (0%)	50.082 | 49.877 (0%)
CSTime (ms)	0.541 | 0.541 (0%)	0.112 | 0.112 (0%)
TexUnitBusy (%)	23.408 | 23.485 (0%)	25.115 | 24.745 (-1%)
CSThreadGroups	2560 | 2560 (0%)	2040 | 2040 (0%)
CSVALUInsts	173.222 | 172.912 (0%)	13.855 | 13.859 (0%)
CSVALUUtilization (%)	38.251 | 38.29 (0%)	82.709 | 83.241 (0%)
CSVALUBusy (%)	30.207 | 30.166 (0%)	7.056 | 7.044 (0%)
CSSALUBusy (%)	3.004 | 2.996 (0%)	5.586 | 5.635 (0%)
CSMemUnitBusy (%)	23.552 | 23.623 (0%)	23.971 | 24.659 (+2%)
CSMemUnitStalled (%)	7.498 | 7.576 (+1%)	7.103 | 6.232 (-12%)
CSFetchSize (kb)	7669.13 | 7667.75 (0%)	8100.69 | 8100.69 (0%)
CSWriteSize (kb)	9177.53 | 9143.88 (0%)	3955.38 | 3944.5 (0%)
CSCacheHit (%)	92.556 | 92.539 (0%)	50.132 | 50.13 (0%)
CSWriteUnitStalled (%)	0.001 | 0 (-100%)	31.943 | 32.606 (+2%)
CSGDSInsts	0 | 0 (0%)	0 | 0 (0%)
CSLDSInsts	8.092 | 8.113 (0%)	0.739 | 0.722 (-2%)
CSALUStalledByLDS (%)	0.001 | 0.001 (0%)	0 | 0 (-10%)
CSLDSBankConflict (%)	0.662 | 0.227 (-65%)	0 | 0 (0%)

HDAO TECHNIQUE

INTEGRATION GUIDE

INTEGRATION

MINIMUM AMOUNT OF CODE FOR INITIAL INTEGRATION

```
extern "C"
{
    AMD_AO_DLL_API AO_RETURN_CODE AO_Initialize          (const AO_DESC & desc);
    AMD_AO_DLL_API AO_RETURN_CODE AO_Render              (const AO_DESC & desc);
    AMD_AO_DLL_API AO_RETURN_CODE AO_Resize              (const AO_DESC & desc);
    AMD_AO_DLL_API AO_RETURN_CODE AO_ChangeLowResolution(const AO_DESC & desc);
    AMD_AO_DLL_API AO_RETURN_CODE AO_Release             (const AO_DESC & desc);
}
```

```
g_AO_DESC.m_pDevice        = pd3dDevice;
g_AO_DESC.m_pDeviceContext = NULL;
AO_Initialize(g_AO_DESC);
```

```
g_AO_DESC.m_pDevice        = DXUTGetD3D11Device();
g_AO_DESC.m_pDeviceContext = DXUTGetD3D11DeviceContext();
g_AO_DESC.m_SrvSize.x      = (float)pBackBufferSurfaceDesc->Width;
g_AO_DESC.m_SrvSize.y      = (float)pBackBufferSurfaceDesc->Height;

g_AO_DESC.m_Camera.m_Fov = g_ViewerCamera.GetFOV();
g_AO_DESC.m_Camera.m_Aspect = g_ViewerCamera.GetAspect();
g_AO_DESC.m_Camera.m_NearPlane = g_ViewerCamera.GetNearClip();
g_AO_DESC.m_Camera.m_FarPlane = g_ViewerCamera.GetFarClip();

AO_Resize(g_AO_DESC);
```

```
TIMER_Begin(0, L"AMD_AO");
{
    g_AO_DESC.m_Camera.m_Fov = g_ViewerCamera.GetFOV();
    g_AO_DESC.m_Camera.m_Aspect = g_ViewerCamera.GetAspect();
    g_AO_DESC.m_Camera.m_NearPlane = g_ViewerCamera.GetNearClip();
    g_AO_DESC.m_Camera.m_FarPlane = g_ViewerCamera.GetFarClip();

    g_AO_DESC.m_SrvSize.x = (float)DXUTGetDXGIBackBufferSurfaceDesc()->Width;
    g_AO_DESC.m_SrvSize.y = (float)DXUTGetDXGIBackBufferSurfaceDesc()->Height;

    g_AO_DESC.m_EnableVerboseLogging = false;
    g_AO_DESC.m_pSrvDepth = g_AppDepth._srv;

    if ( g_HUD.m_GUI.GetCheckBox( IDC_CHECKBOX_AO_ENABLE_NORMALS )->GetChecked())
    {
        g_AO_DESC.m_pSrvNormal = g_AppNormal._srv;
        g_AO_DESC.m_NormalOption = AMD::AO_NORMAL_OPTION_READ_FROM_NORMAL_SRV;
    }
    else
    {
        g_AO_DESC.m_pSrvNormal = NULL;
        g_AO_DESC.m_NormalOption = AMD::AO_NORMAL_OPTION_NONE;
    }

    g_AO_DESC.m_pRtvOutputAO = g_AOResult._rtv;
    g_AO_DESC.m_pDeviceContext = pd3dContext;
    g_AO_DESC.m_OutputChannelsFlag = 8|4|2|1;
    AO_Render(g_AO_DESC);

    g_AO_DESC.m_EnableCapture = false;
}
TIMER_End();
```

DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION