Rendering in Codemasters' GRID2 and beyond: Achieving the ultimate graphics on both PC and tablet

Richard Kettlewell, Codemasters

Leigh Davies, Intel

Legal

*Other names and brands may be claimed as the property of others.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED,
BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS
OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Any code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel product plans in this presentation do not constitute Intel plan of record product roadmaps. Please contact your Intel representative to obtain Intel’s current plan of record product roadmaps.

Performance claims: Software and workloads used in performance tests may have been optimized for performance only on Intel® microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to
http://www.Intel.com/performance

Iris™ graphics is available on select systems. Consult your system manufacturer.

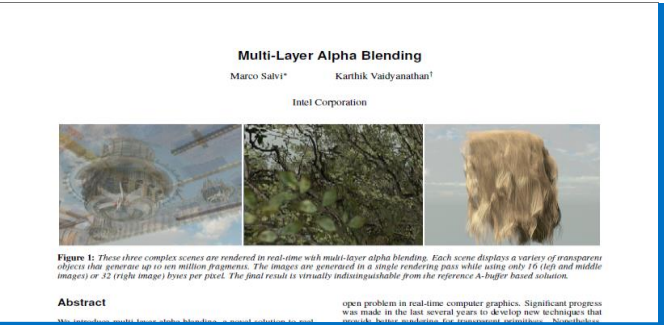
Intel, Intel Inside, the Intel logo, Intel Core and Iris are trademarks of Intel Corporation in the United States and other countries.

We are covering a lot:- Don't panic
More resources are available online☺

http://software.intel.com/en-us/vcsource/samples

http://software.intel.com/en-us/vcsource/learn/techarticles

http://software.intel.com/en-us/siggraph2013


Multi-Layer Alpha Blending

Marco Salvi* Karthik Vaidyanathan†

Intel Corporation

Figure 1: These three complex scenes are rendered in real-time with multi-layer alpha blending. Each scene displays a variety of transparent objects that generate up to ten million fragments. The images are generated in a single rendering pass while using only 16 (left and middle images) or 32 (right image) bytes per pixel. The final result is virtually indistinguishable from the reference A-buffer based solution.

Abstract

open problem in real-time computer graphics. Significant progress was made in the last several years to develop new techniques that

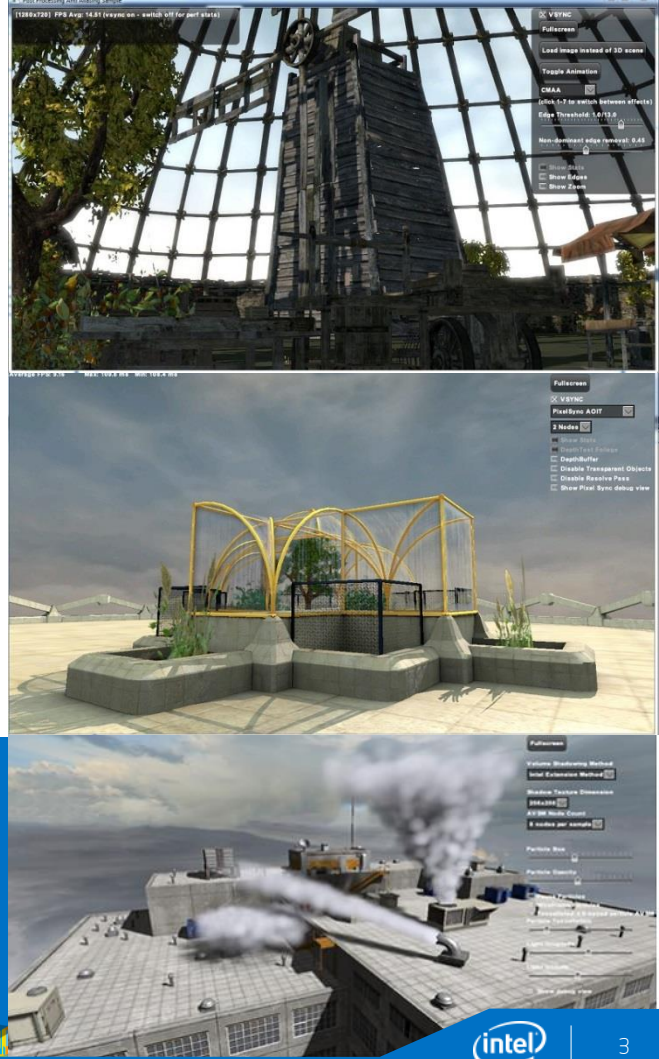

intel Look Inside.

From Research to Production

How AVSM and AOIT made their way into GRID 2

Richard Kettlewell, Codemasters

Axel Mamode, Intel



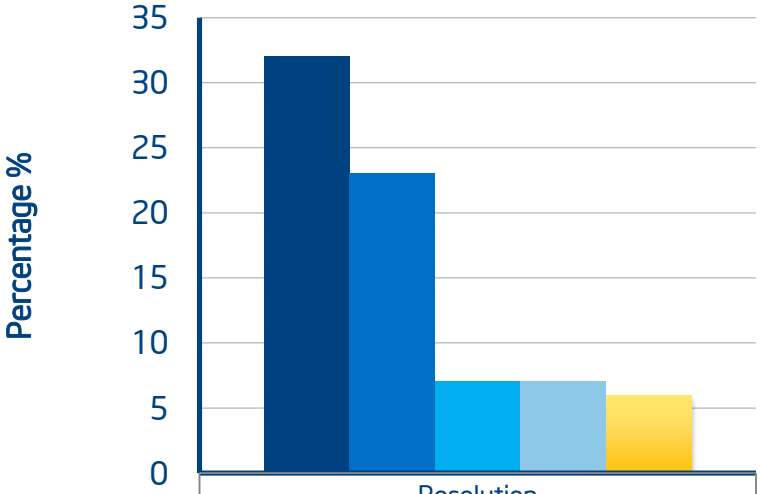
Codemasters: 28 Years of Cross Platform AAA Development

Differentiate PC through cutting edge technology

- Multi-threading (2007)
- Early mover on DX11 (2009)
  - Tessellation, Compute
- Forward+ Lighting (2012)

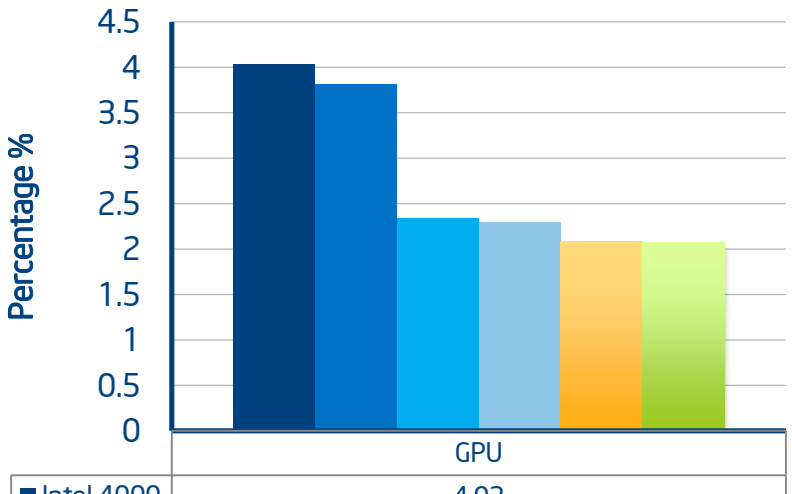
The world is Changing

Common PC Resolutions



Resolution	
1920x1080	32
1366x768	23
1600 x 900	7
1280 x 1024	7
1440 x 900	6

Most popular PC GPU's

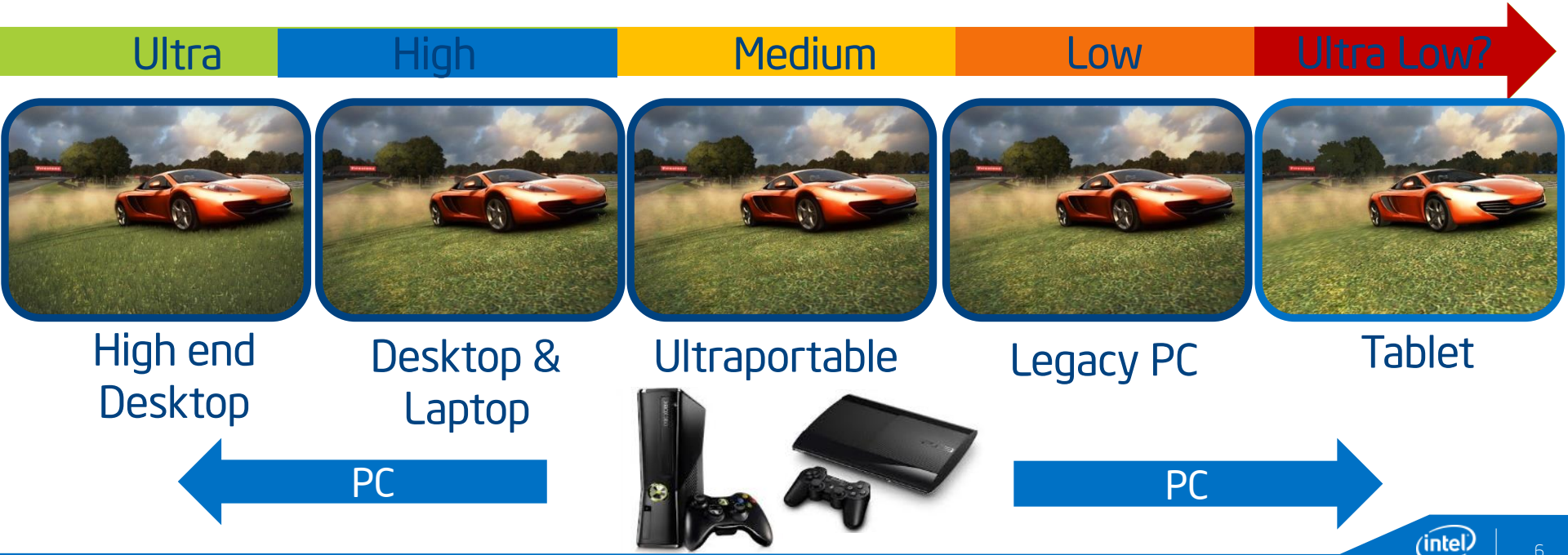


GPU	
Intel 4000	4.03
Intel 3000	3.81
Intel 2000	2.34
GTX 660	2.29
GTX 550TI	2.08
GTX 560TI	2.07

Source Steam Power Hardware survey Jan 2014

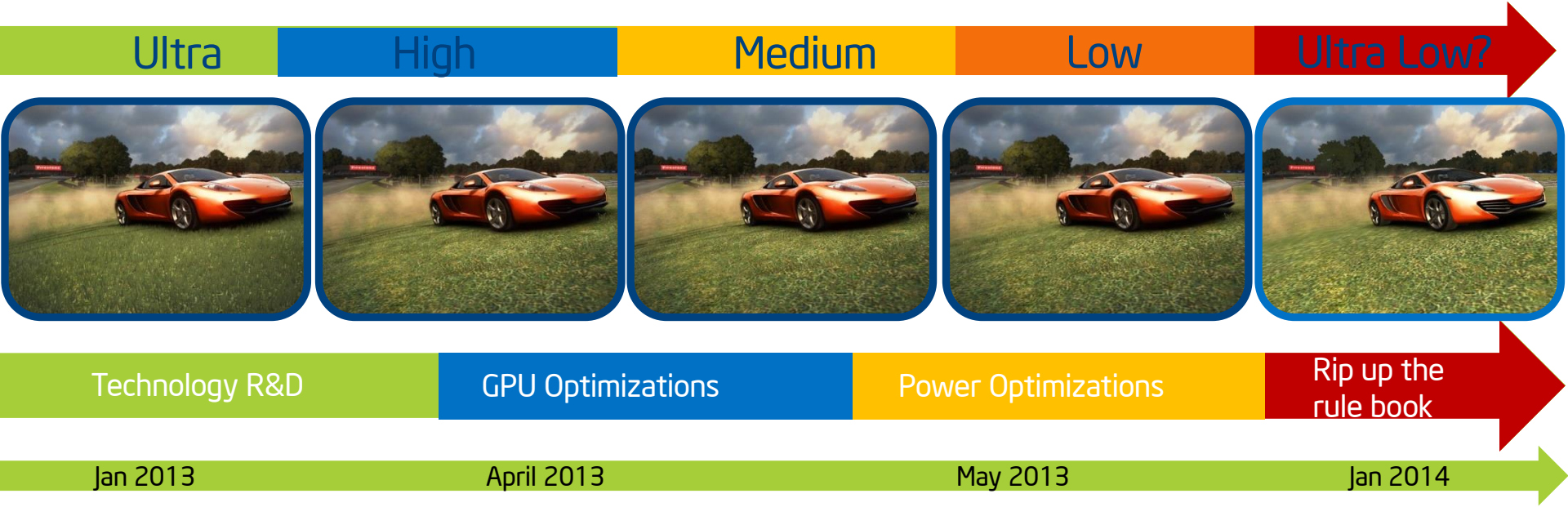
Goals in 2013/14?

- Make medium settings match current consoles
- Scale up and down from console quality



Goals in 2013/14?

- Make medium settings match current consoles
- Scale up and down from console quality



Pixel Shader Ordering (Making the impossible possible!)

What is it?

- Pixel Shader Mutex at a screen location
- Fast as only colliding threads are serialized
- Guaranteed Execution Order

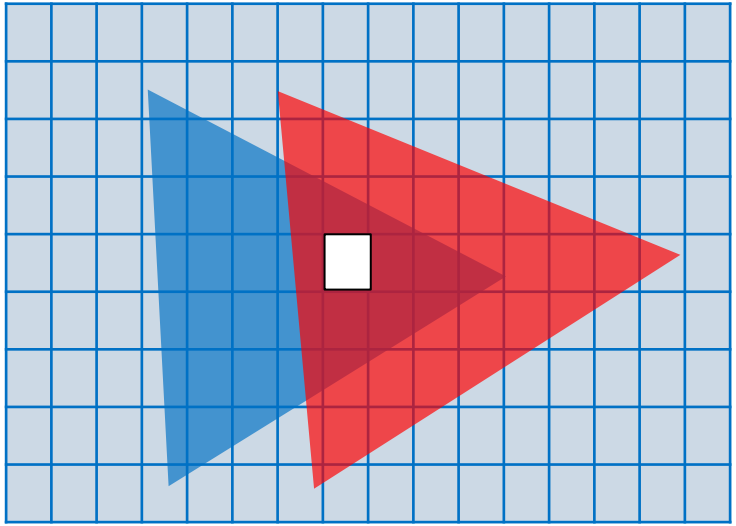
Similar to Alpha Blending rules

- Pixels written in **SV_PrimitiveID** order
- Pixel Shader Ordering moves this guaranteed ordering into the Pixel Shader

What can we use this for?

- Anything that wants to read-modify-write a per-pixel data structure

Introduced with 4th Generation Intel Core Processors



Overlapping pixels can execute in parallel without Pixel Shader Ordering

Pixel Shader Ordering (Making the impossible possible!)

What is it?

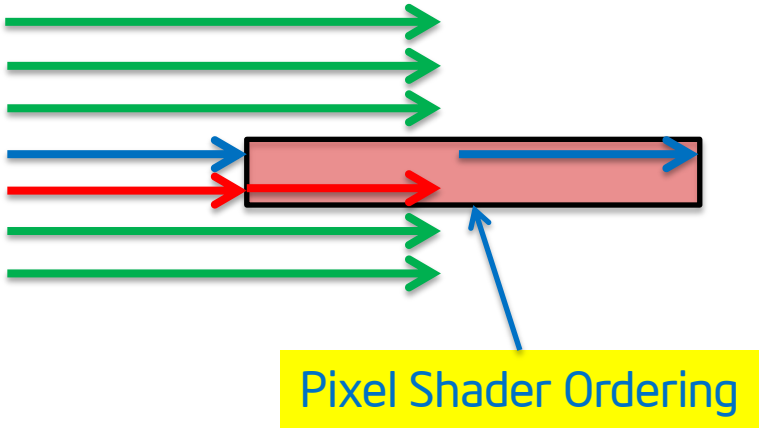
- Pixel Shader Mutex at a screen location
- Fast as only colliding threads are serialized
- Guaranteed Execution Order

Similar to Alpha Blending rules

- Pixels written in **SV_PrimitiveID** order
- Pixel Shader Ordering moves this guaranteed ordering into the Pixel Shader

What can we use this for?

- Anything that wants to read-modify-write a per-pixel data structure



Technology R&D
How AVSM and OIT made their way into GRID 2
Sneak peek at programmable blending

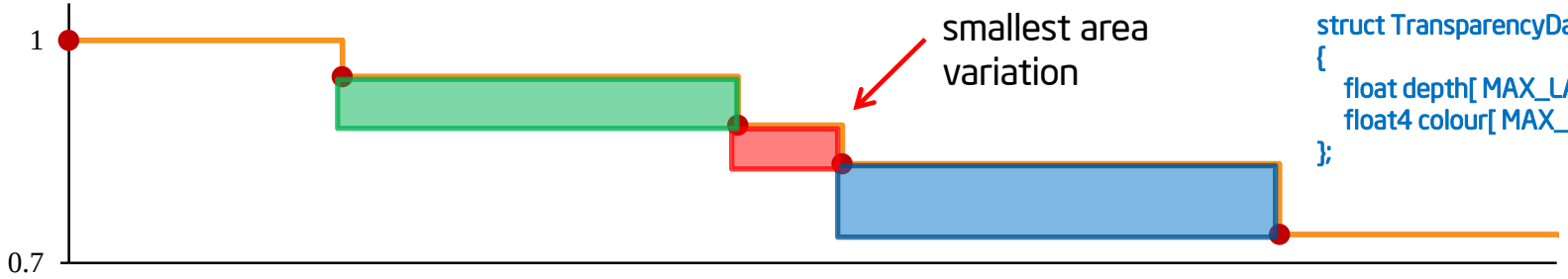
OIT (Order Independent Transparency)

- Represent multiple layers of transparency without sorting problems
- Denser/softer looking foliage (especially in distance due to mips)
- Improved other alpha tested geometry

Store Visibility Function as a sorted fixed-size array of nodes, in UAV surface

Each ***red*** node corresponds to a pair of values for depth and transmittance: (d,t)

To compress visibility we remove the node that generates the smallest area variation

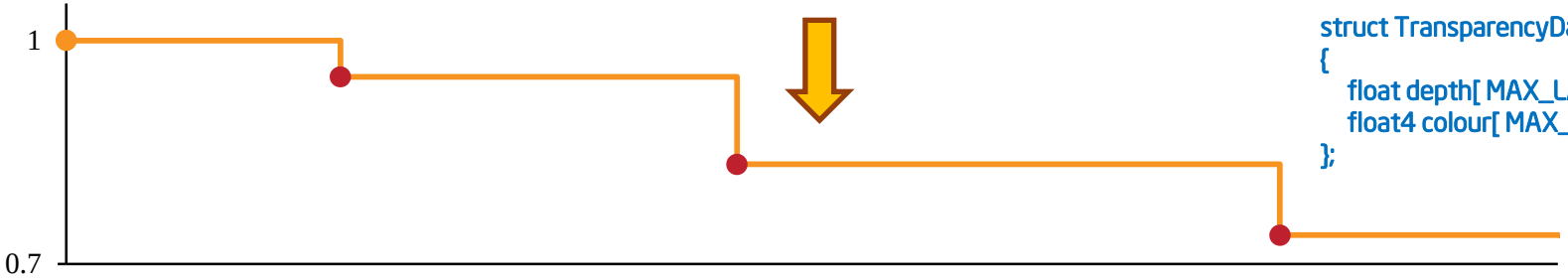



```
struct TransparencyData
{
    float depth[ MAX_LAYERS ];
    float4 colour[ MAX_LAYERS ];
};
```

Store Visibility Function as a sorted fixed-size array of nodes, in UAV surface

Each ***red*** node corresponds to a pair of values for depth and transmittance: (d,t)

To compress visibility we remove the node that generates the smallest area variation




```
struct TransparencyData
{
    float depth[ MAX_LAYERS ];
    float4 colour[ MAX_LAYERS ];
};
```

final_color = $\sum c_i \alpha_i vis(z_i)$

Final full screen resolve to composite OIT data with main image

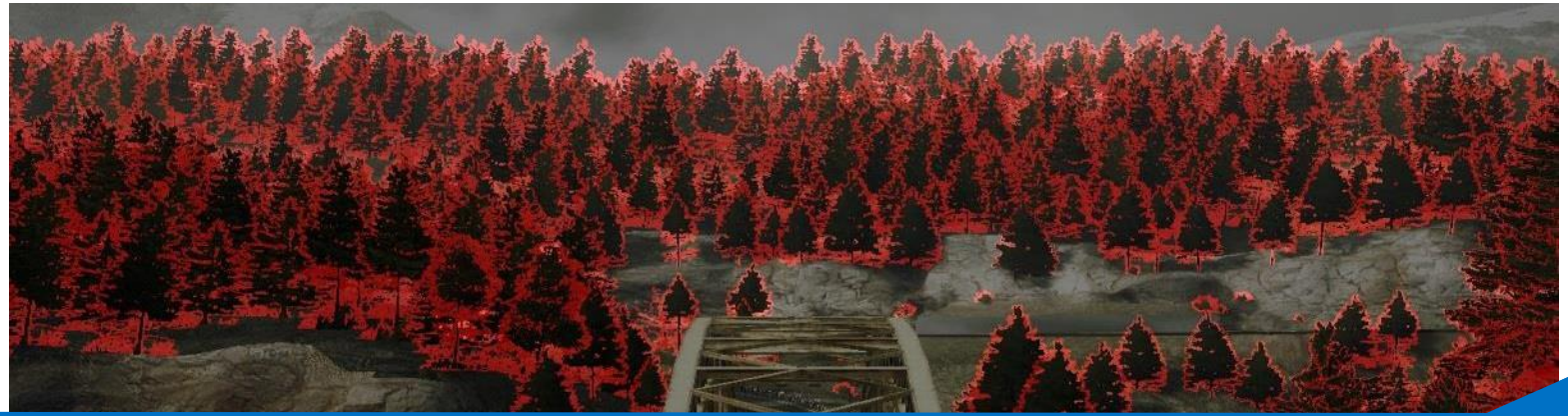
Alpha Coverage in GRID2

Lot of semi-transparency in foliage (See Red Mask)

Alpha blending isn't an option....

Original system used Alpha 2 Coverage, but requires 4xMSAA to look good.

Without Blending or A2C we get aliased results



Draw Order Problems?

Must composite full screen quad amongst other transparencies

Result of not doing full scene OIT

Organise draw order to solve most problems

God rays/Haze required a different approach

Draw Order Problems?

Must composite full screen quad amongst other transparencies

Result of not doing full scene OIT

Organise draw order to solve most problems

God rays/Haze required a different approach

Draw Order Solution

Can't put god ray polygons into OIT

- They are far too big!
- Only need OIT on the areas that overlap

Solution: Only add haze/god-ray pixels into the OIT if there is already tree OIT data in the buffer

- Store mask of pixels in R32 target to flag pixels that contain OIT data
- Use the mask to identify pixels where the haze/god rays need to be rendered with OIT
- All other pixels are rendered with normal alpha blending, because there is no overlap

OIT Performance

2-node OIT

Tiled memory access

- *(y * width) + x* is bad!

Clear mask texture

Worst case of 3.1ms (2.5ms + 0.6ms)*

Typical case of around 2ms*

* On a 4th Gen Core™ Processor with Intel® Iris™ Pro Graphics, at 1600x900

AVSM

- Adaptive Volumetric Shadow Mapping
- Can we use a similar idea to approximate light transmittance through participating media?
- Render OIT from the light's point of view
- Can be used to render volumetric smoke effects, such as tyre kickup

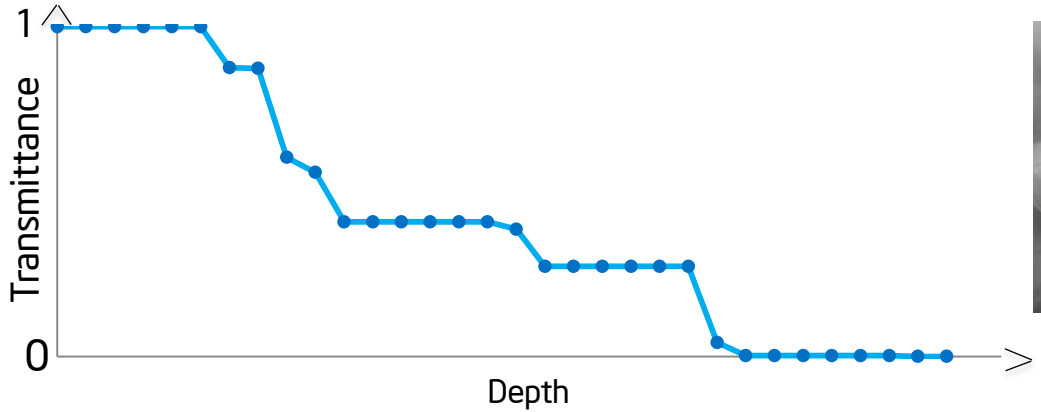
AVSM – Problem Background

Realistic lighting of volumetric media

- Hair, smoke, fog, etc..

Compute visibility curve

- **Transmittance**: Fraction of light that passes through a material

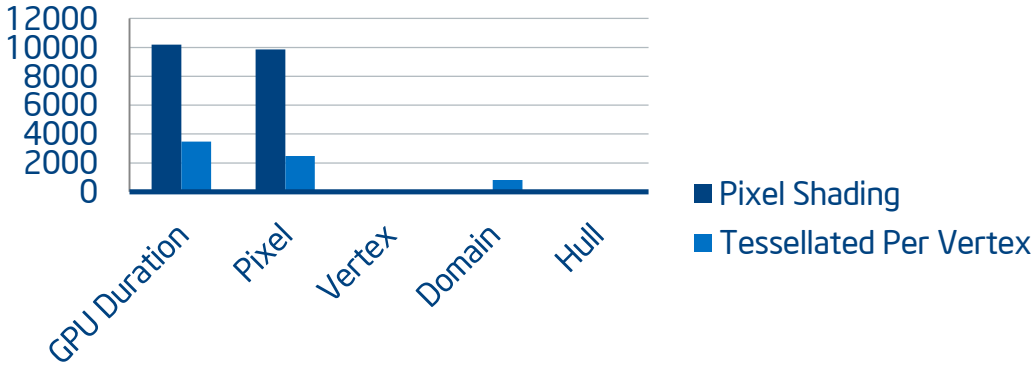


Original Tyre Smoke

AVSM Tyre Smoke

Lighting particles

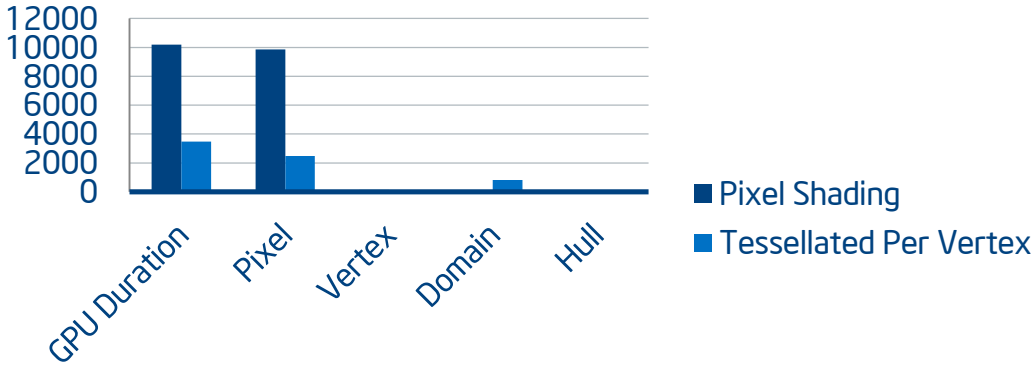
Original R&D focused on optimizing shadow map Read/Write



Per Pixel lighting of particles took >=10ms...

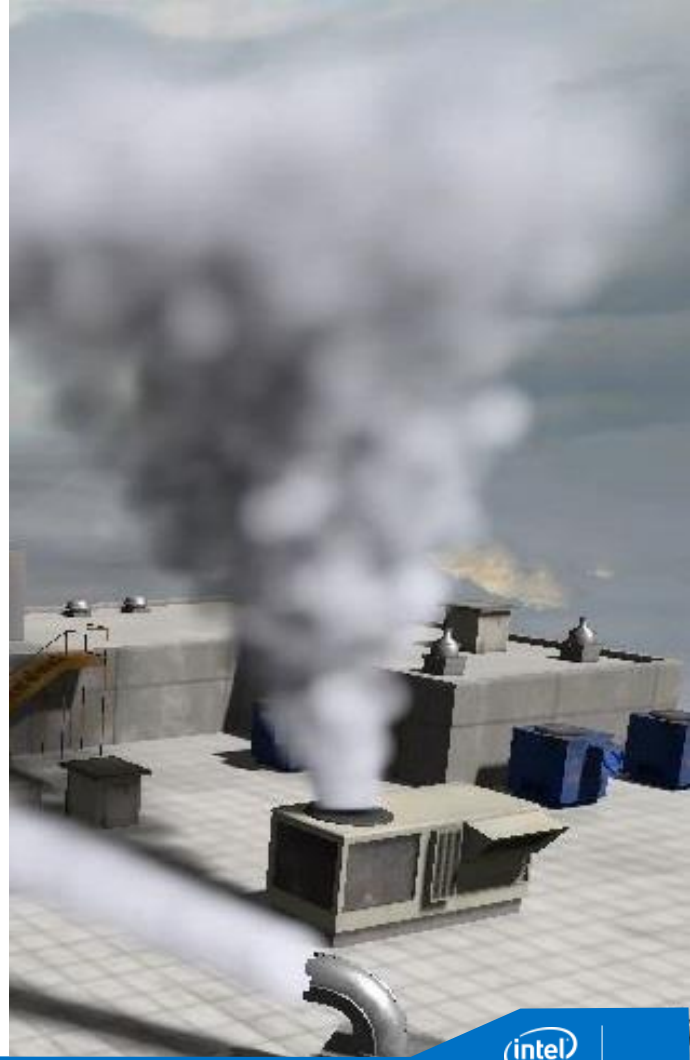
Lighting particles

Original R&D focused on optimizing shadow map Read/Write



Per Pixel lighting of particles took >=10ms...

- Per vertex lighting is too coarse.
- Per vertex with screen space tessellation actually looked better!
- Tessellated per vertex is 2-3x faster



Particle Sorting

Problem: Overlapping emitters weren't sorting correctly

Idea: Use OIT!

- Requires high node count (16 nodes!)
- Uses precious GPU performance

CPU sorting?

- Facing billboards makes CPU sorting possible
- We had spare performance on the CPU

Particle Sorting

Problem: Overlapping emitters weren't sorting correctly

Idea: Use OIT!

- Requires high node count (16 nodes!)
- Uses precious GPU performance

CPU sorting?

- Facing billboards makes CPU sorting possible
- We had spare performance on the CPU

AVSM Performance

Actual worst case of 15ms (9ms + 6ms)*

Typical worst case of 4.5ms (2.5ms + 2ms)*

Average case much lower (2ms for entire effect)

* On a 4th Gen Core™ Processor with Intel® Iris™ Pro Graphics, at 1600x900

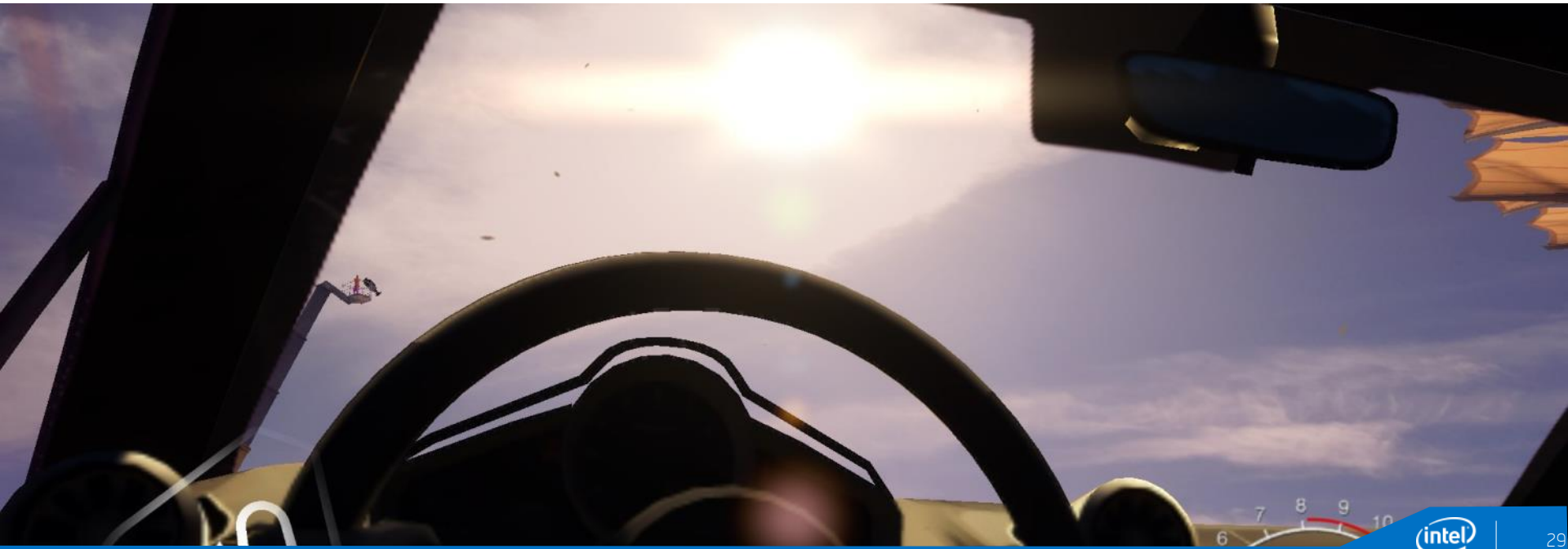

889
intel

Programmable Blending

- HDR lighting values encoded logarithmically into R10G10B10A2 back buffer
- Fixed function alpha blending of encoded values is invalid
- Result is loss of high dynamic range behind transparencies
- Solution is to blend in linear space

Programmable Blending

- HDR lighting values encoded logarithmically into R10G10B10A2 back buffer
- Fixed function alpha blending of encoded values is invalid
- Result is loss of high dynamic range behind transparencies
- Solution is to blend in linear space



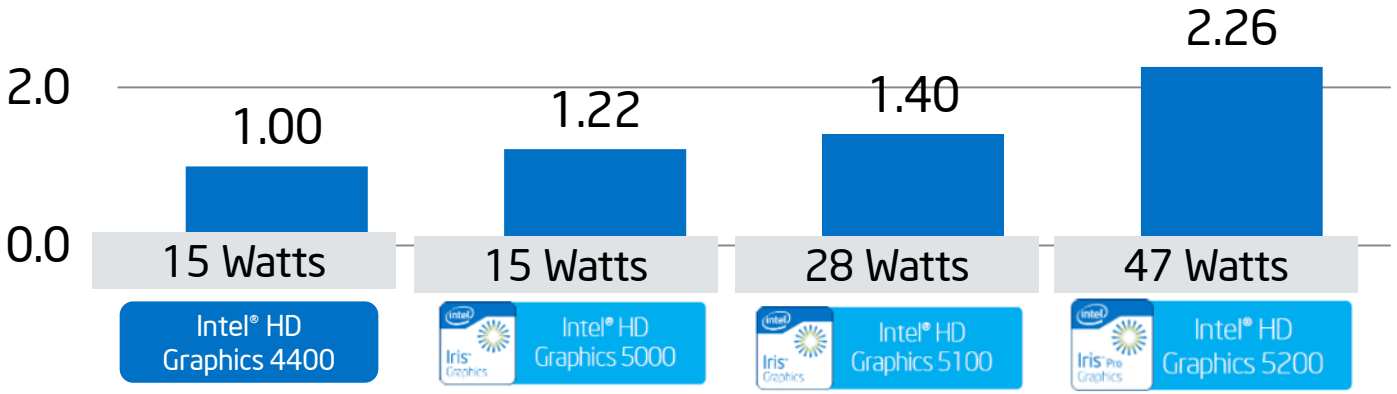
GPU and CPU Optimizations:

- Different to optimizing a system using discrete graphics
- Some optimizations were counter intuitive
- Optimizing for Power and Bandwidth played a big part in getting expected performance

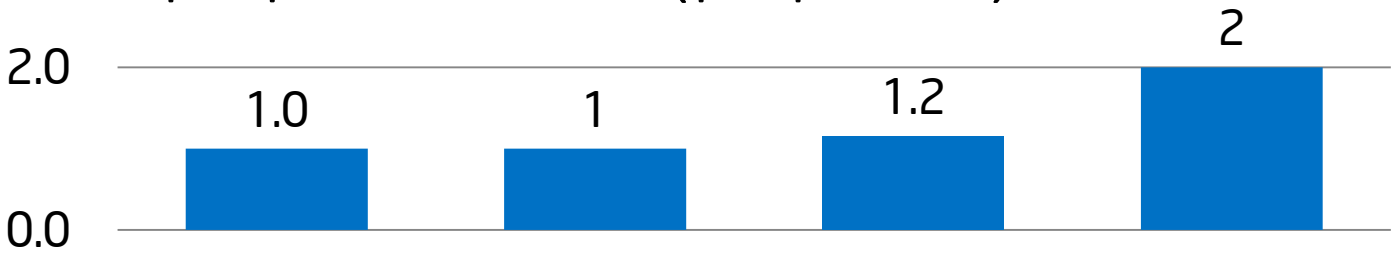
*

Performance scaling

Relative Graphics benchmark performance

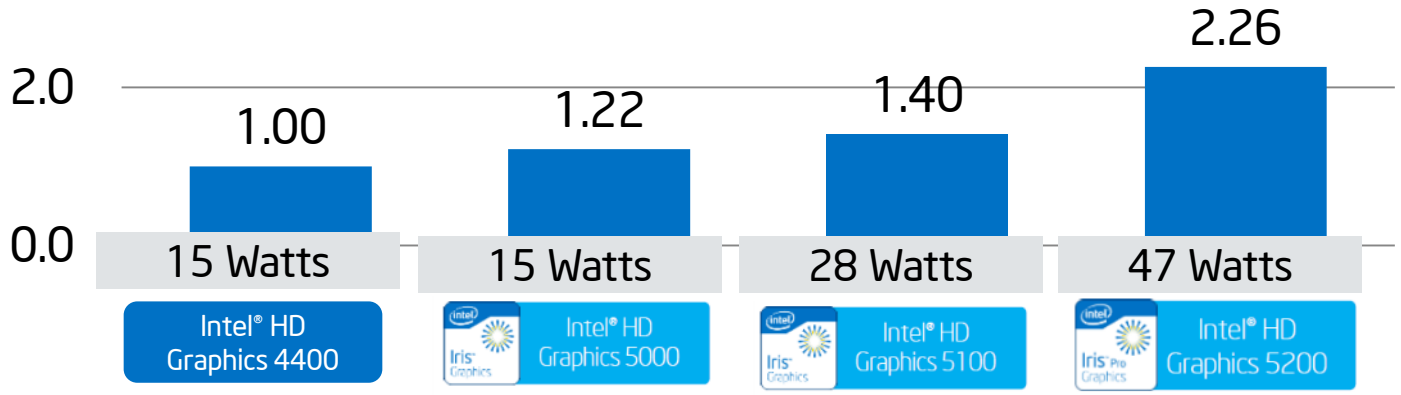


Initial Graphics performance in GRID2 (pre-optimization)

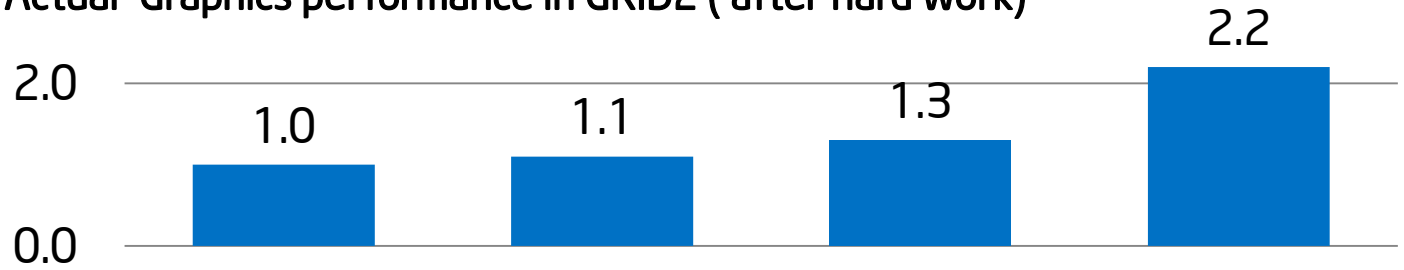


Performance scaling

Relative Graphics benchmark performance



Actual Graphics performance in GRID2 (after hard work)



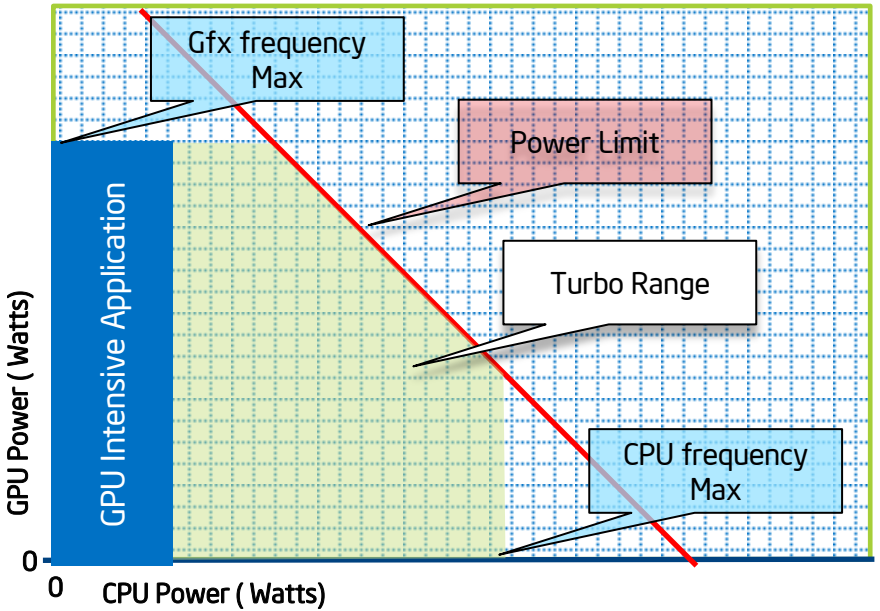
Why?

Welcome to the world of power sharing

CPU and GPU share the Thermal Design Power (TDP) rating for the system.

CPU and GPU have maximum allowed frequencies, you get one or the other, not both at the same time!

In graphics benchmarks the load sharing looks like this

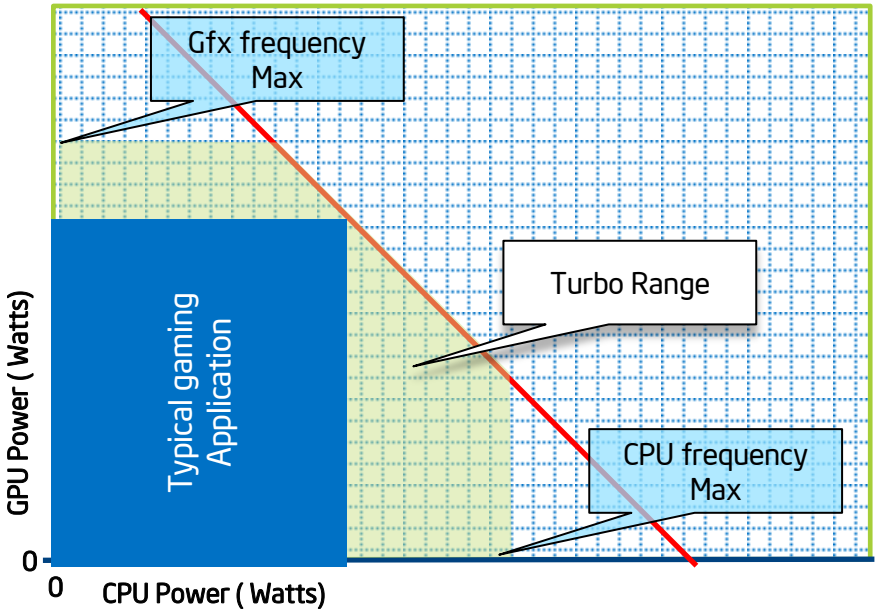


Games look more like this!

TDP shared more evenly between CPU and GPU.

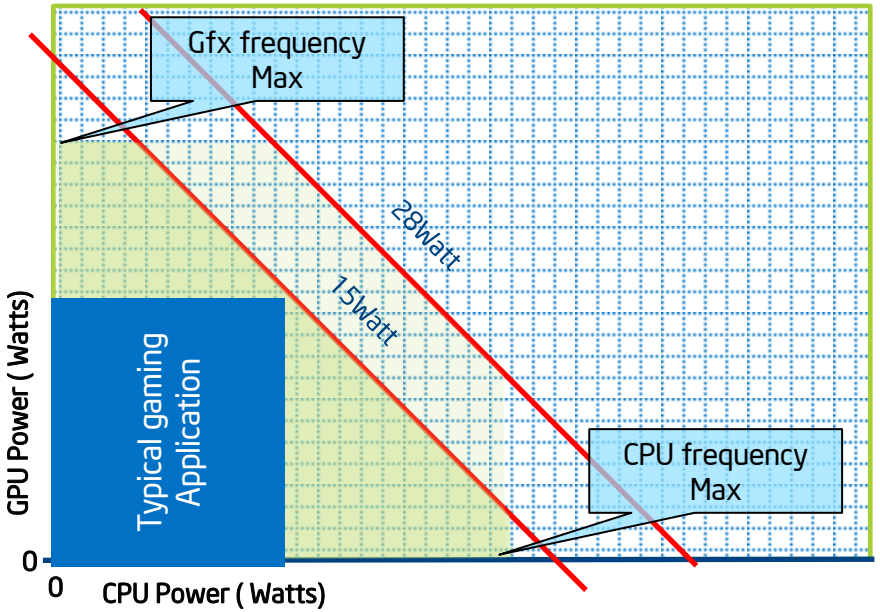
Audio, AI, higher graphics API overheads all lead to higher CPU usage.

Higher CPU requirements means can be hard to hit Max Gfx frequency.



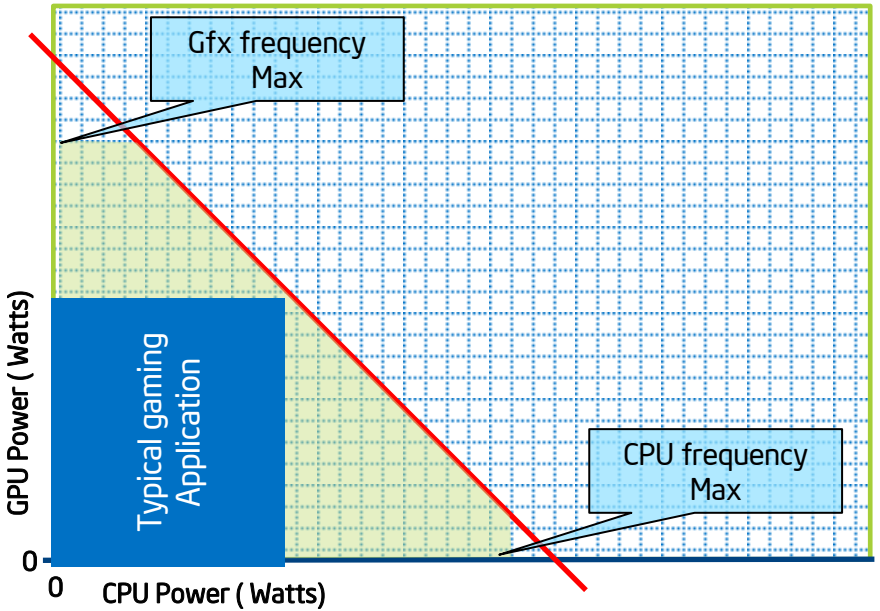
Lower the TDP, more aggressive trade-off

Max CPU and GPU frequency might not change much at lower TDP, but you can't get both at the same time



So What happens when you optimize graphics?

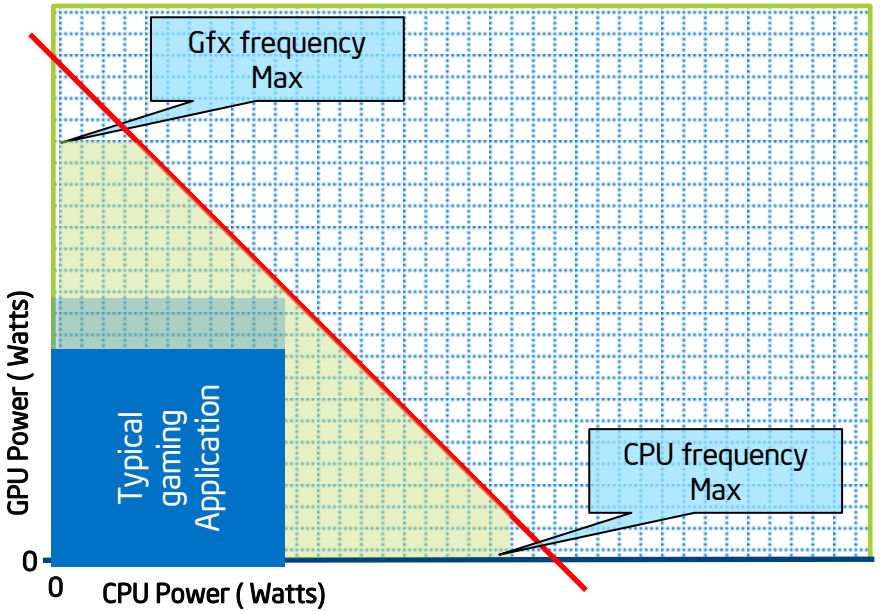
If profiling tells you that your GPU bound, then optimizing the GPU will improve performance right????



So What happens when you optimize graphics?

If profiling tells you that your GPU bound, then optimizing the GPU will improve performance right????

You have a great day and save 20% of your GPU workload☺ Do you get 20% extra FPS??

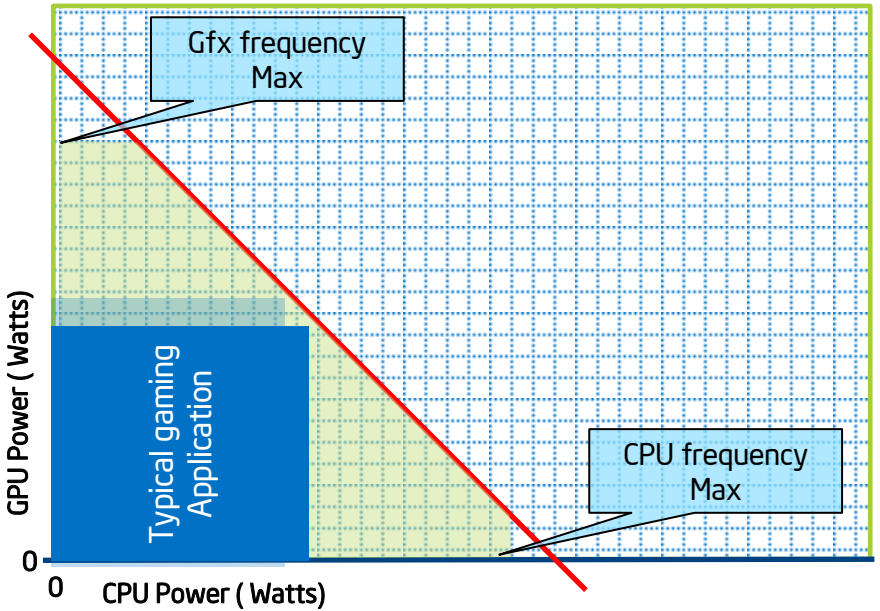


So What happens when you optimize graphics?

If profiling tells you that your GPU bound, then optimizing the GPU will improve performance right????

You have a great day and save 20% of your GPU workload☺ Do you get 20% extra FPS??

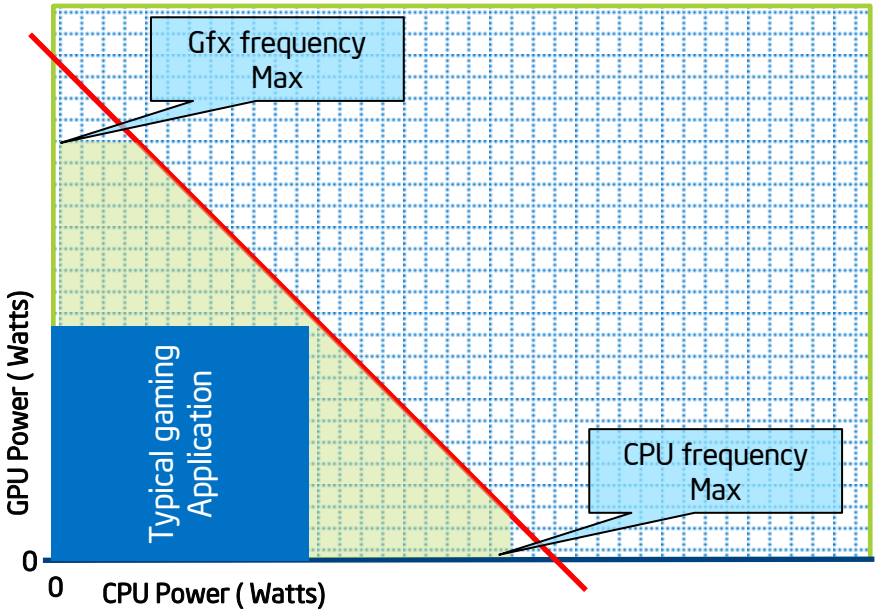
Extra FPS normally requires more CPU to drive the workload.



GPU bound? Optimize the CPU!!

Sounds crazy but increasingly common.

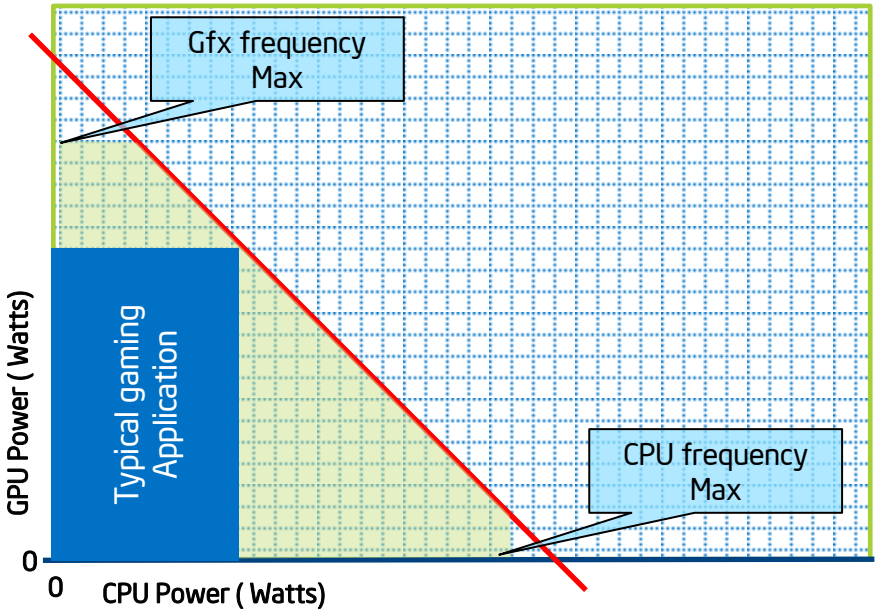
- GPU and CPU share power budget
- Frequencies dynamically adjusted at run time based on workload
- Optimizing one gives more power to the other
- Base CPU frequency can be very misleading…..



GPU bound? Optimize the CPU!!

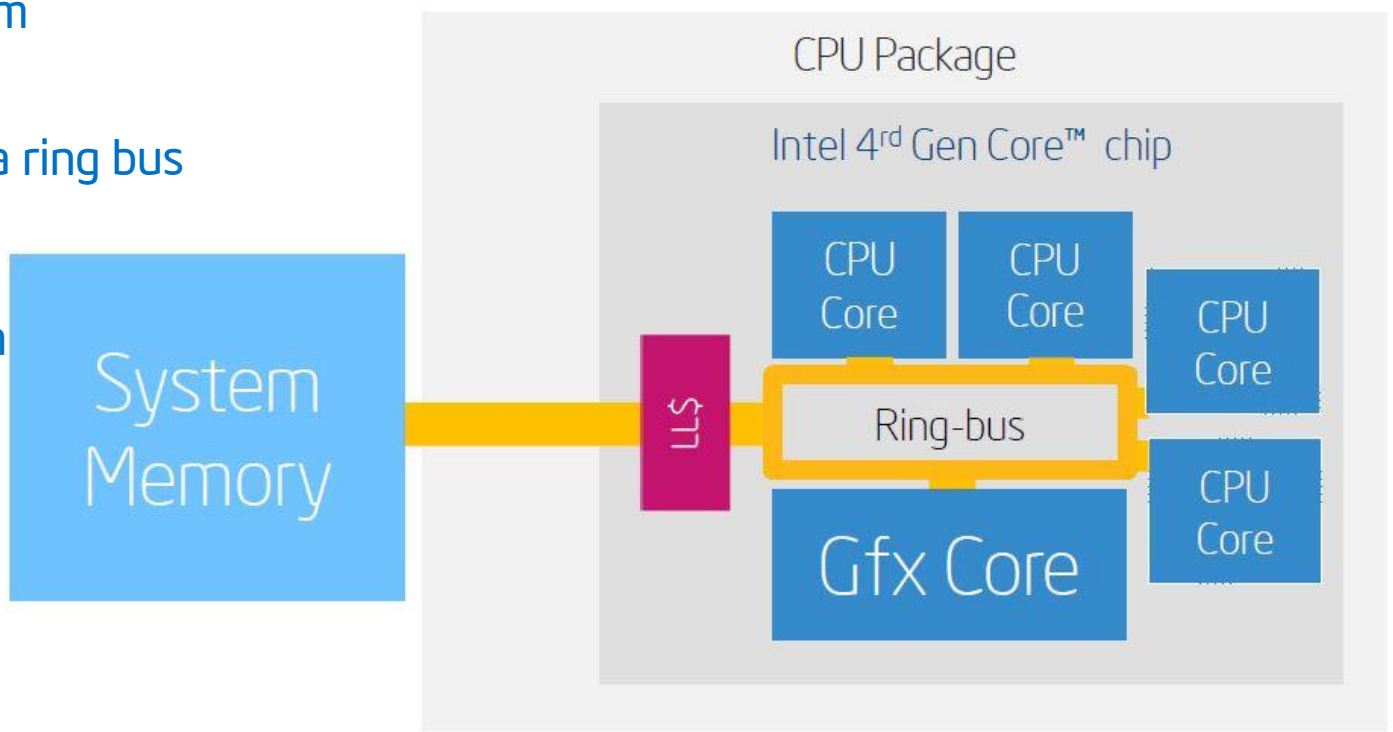
Sounds crazy but increasingly common.

- GPU and CPU share power budget
- Frequencies dynamically adjusted at run time based on workload
- Optimizing one gives more power to the other
- Base CPU frequency can be very misleading…..



Power isn't the only thing shared!

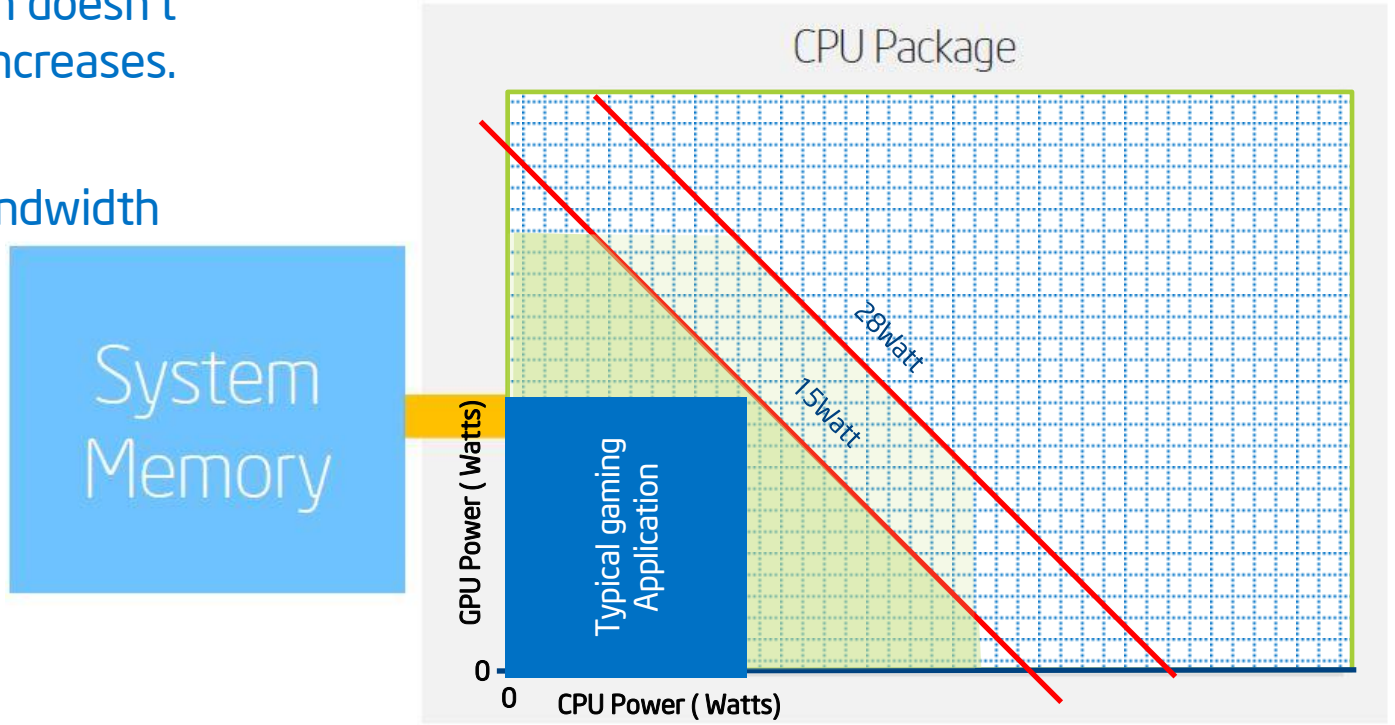
- Up to 1.7Gb of system memory
- Connected to CPU via ring bus
- Shared LL$.
- Bandwidth to system shared between CPU and GPU.



The Juggling Game continues

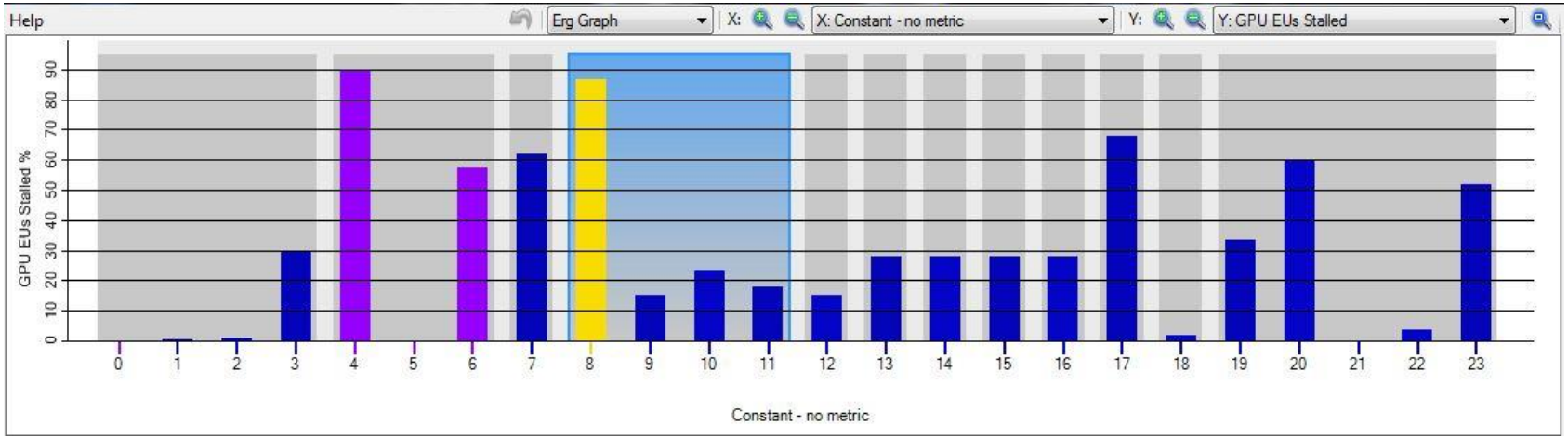
Off package bandwidth doesn't change much as TDP increases.

Increasing GPU or CPU workload increases bandwidth requirements



Can you feed the system fast enough?

- If a higher TDP doesn't give much more performance, check how busy the GPU is.
- EU stalls can often be either directly caused by waiting on RAM, or indirectly via the sampler.
- Can be checked in Intel GPA.



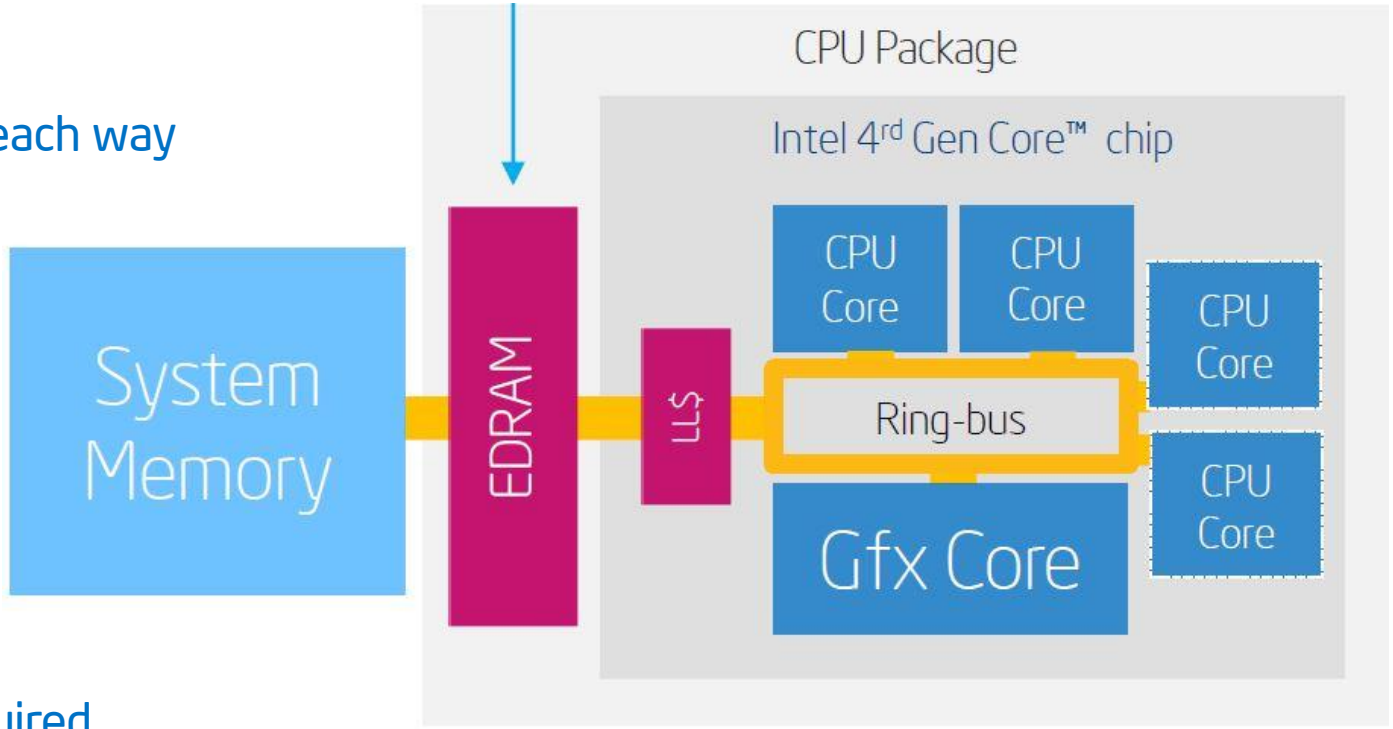
Iris™ Pro

On same package as CPU

128MB

Bandwidth 50GB/sec each way

Acts as 4th level cache



Just works, no API required

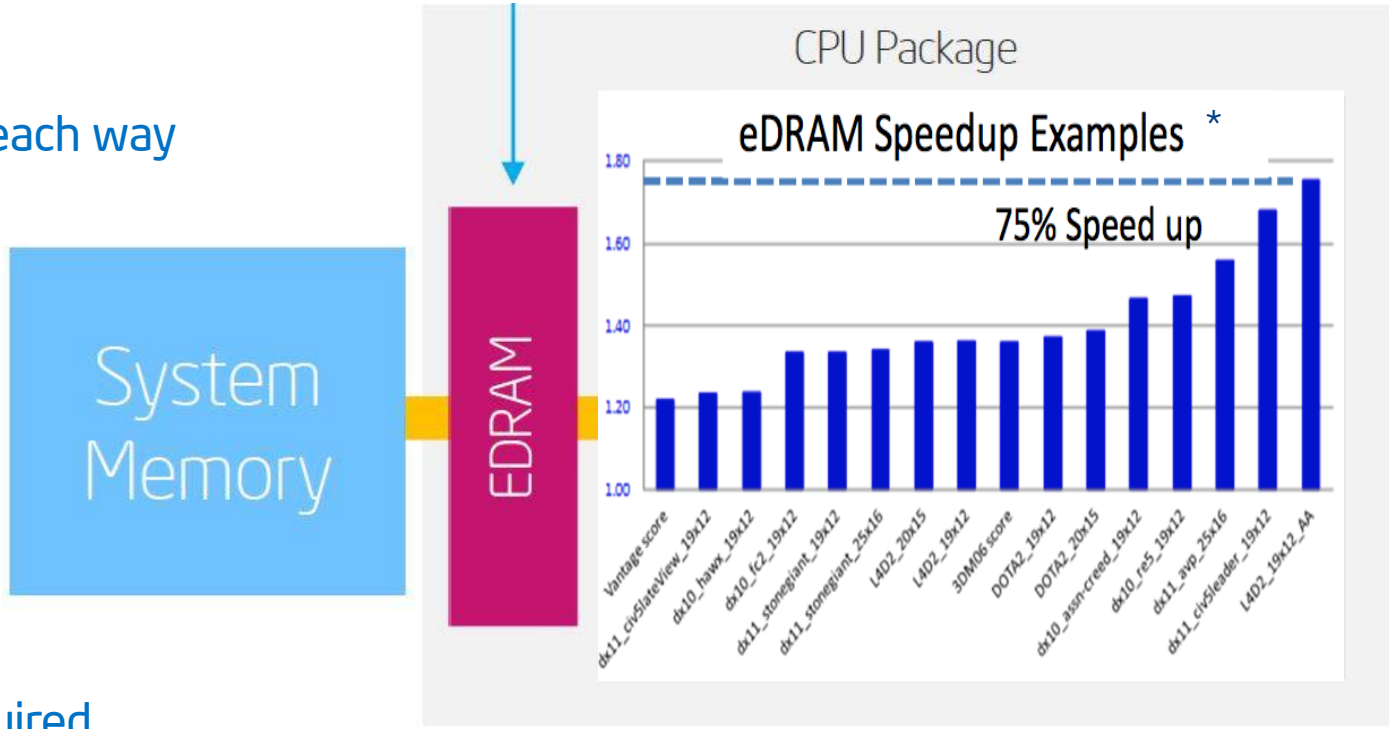
Iris™ Pro

On same package as CPU

128MB

Bandwidth 50GB/sec each way

Acts as 4th level cache



Just works, no API required

* © 2014 IEEE
International Solid-State Circuits Conference

SSAO: When R&D doesn't scale down…

- Disproportionately expensive on medium settings, 15-20% of a frame.
- Was CS based, difficult to optimize across multiple hardware vendors
- Very BW memory intensive
  - 2 depth samples per occlusion result
  - Smart cross bilateral blur reads from depth to determine edges
- Worked at ½ x ½ screen resolution

	IvyBridge Ultrabook	Haswell GT3e SDP*	NVidia GTX 470
OLD, low quality	8.9ms	2.71ms	1.33ms
OLD, high quality	10.8ms	4.2ms	1.78ms

* Pre-production hardware

SSAO: Reinventing The Wheel

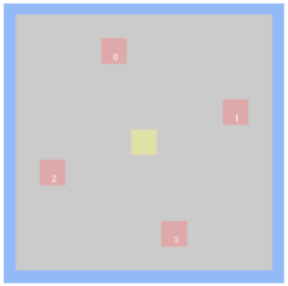
Based on Image-Space Horizon-Based Ambient Occlusion

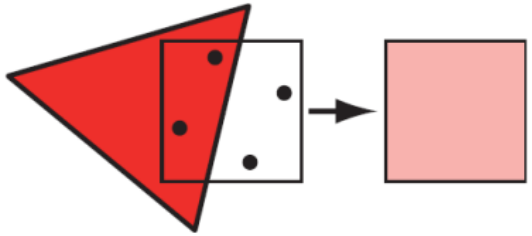
- Completely PS-based, still ½ x ½ res
- Base cost for normal & edge detection + one depth sample for one occlusion result
- Smart cross bilateral blur uses edges from previous pass, doesn't read Depth

	IvyBridge Ultrabook	Haswell GT3e SDP*	NVidia GTX 470
NEW, low quality	2.1ms	0.85ms	0.41ms
NEW, high quality	3.8ms	1.38ms	0.84ms

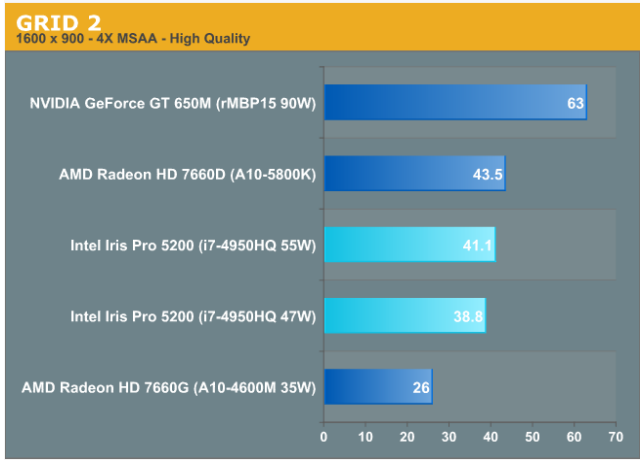
* Pre-production hardware

MSAA Performance





- Pixel shader run once per sample (Yellow dot). Coverage and Occlusion done at higher rates.
- Storage required at a Subsample level, increases bandwidth and memory requirements
- Costs vary on hardware and workload but its never free.



Intel Iris Pro 5200 Graphics Review: Core i7-4950HQ Tested

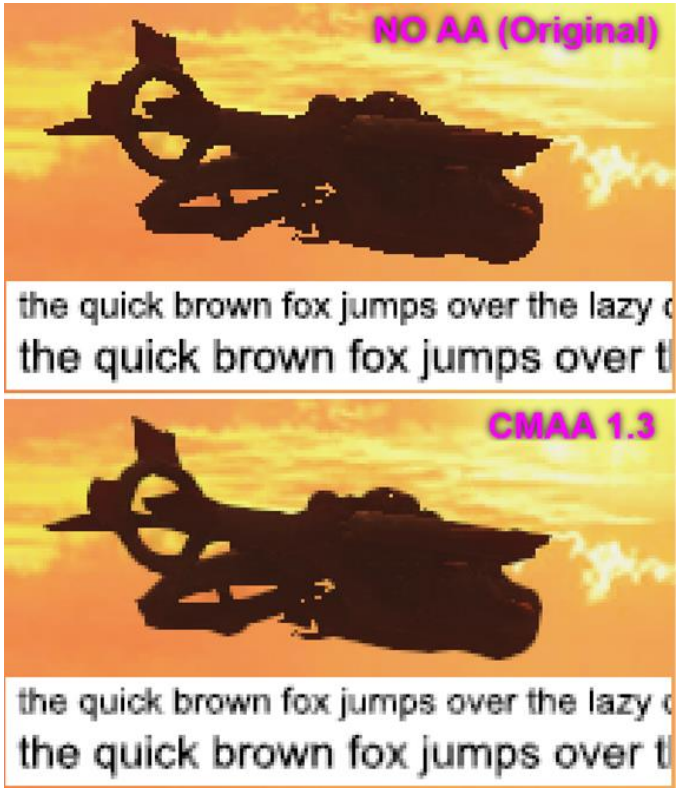
by Anand Lal Shimpi on June 1, 2013 10:01 AM EST

Post-Process AA as an alternative

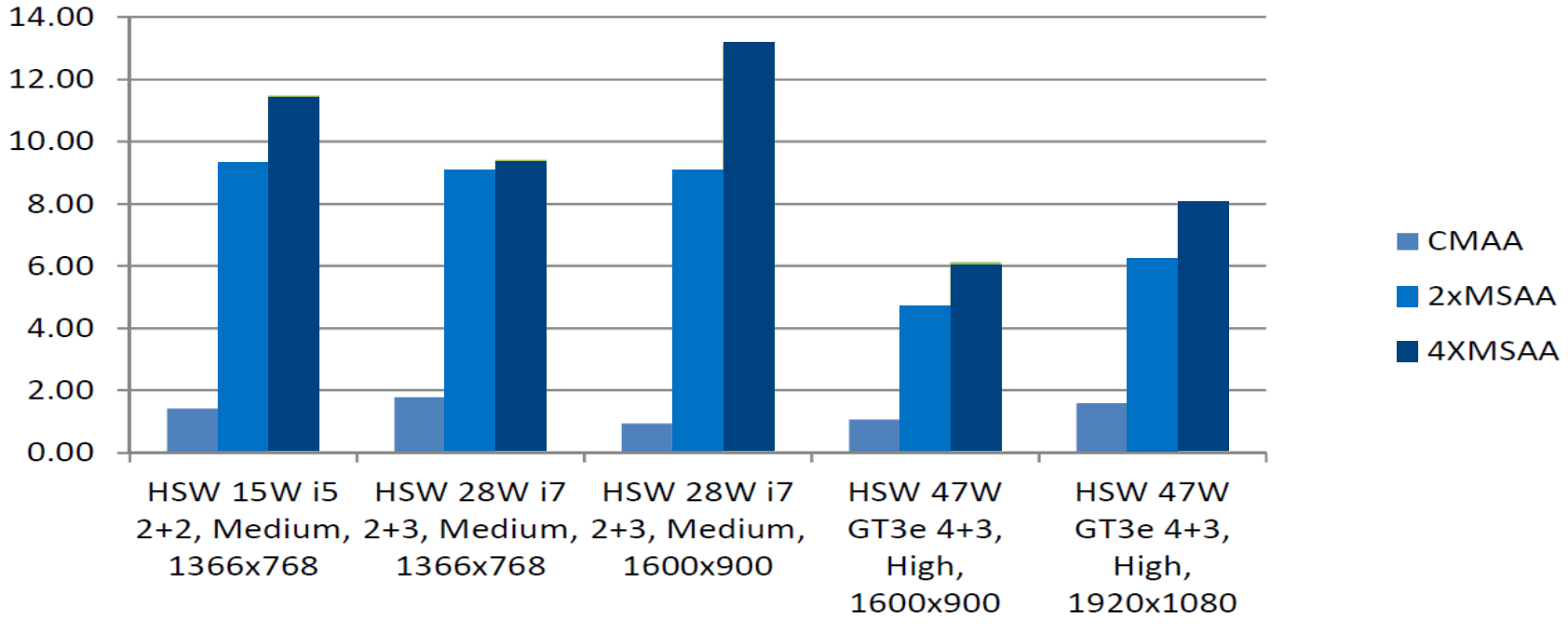
- Evaluated two of the most common ones: SMAA 1x and FXAA 3.11for GRID2
- FXAA 3.11 – good performance but developer found it too blurry: (detail loss on text and high frequency textures)
- SMAA 1x – a bit too costly to beat MSAA for forward rendering, still a bit blurry
- Started with “Morphological Antialiasing” [2009 Alexander Reshetov, Intel Labs]
- Post process that detects aliasing by analyzing colour discontinuities (edges), and applies smart blur to reduce aliasing

Enter Conservative Morphological AA (CMAA)

- Based on MLAA, but solving only symmetrical Z shapes instead of U, Z and L-shapes
- Better preservation of average image colour and temporal stability.
- Conservative approach to determining and pruning edges. "if unsure, don't blur"
- Overall less damaging and higher AA quality compared to FXAA 3.11
- Tailored for Intel Haswell: as fast as FXAA 3.11, twice as fast as SMAA 1x.

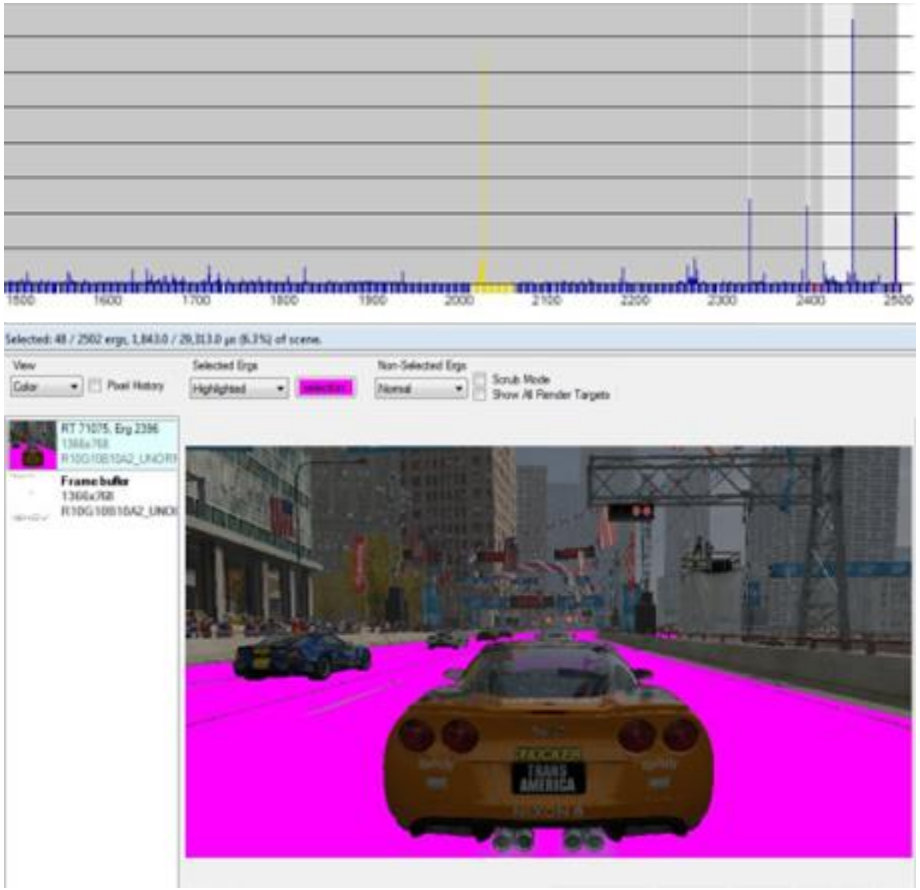


GRID2 performance comparison vs MSAA (milliseconds per frame)



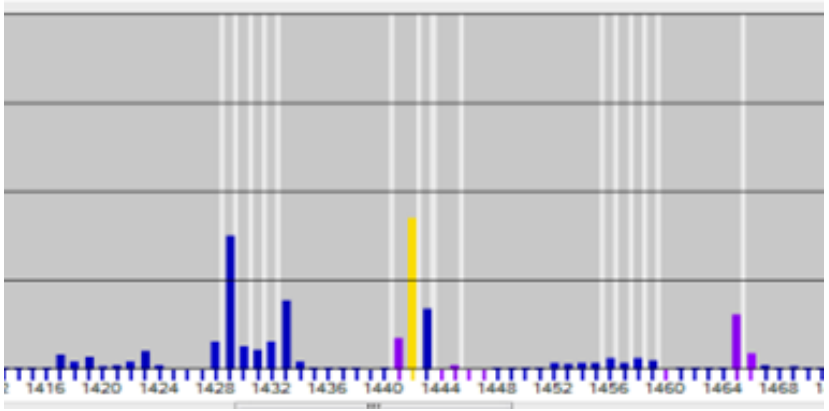
Road Pixel Shader

- Blocked on pixel shaders
  - Eu pixel shader stall = 48.2%
  - Aniso filtering stalls samplers
    - 3.7 pixel cache lines accessed per sample.
- New menu option added to scale artist set anisotropic levels.
  - <20% on medium
  - <5% on low.



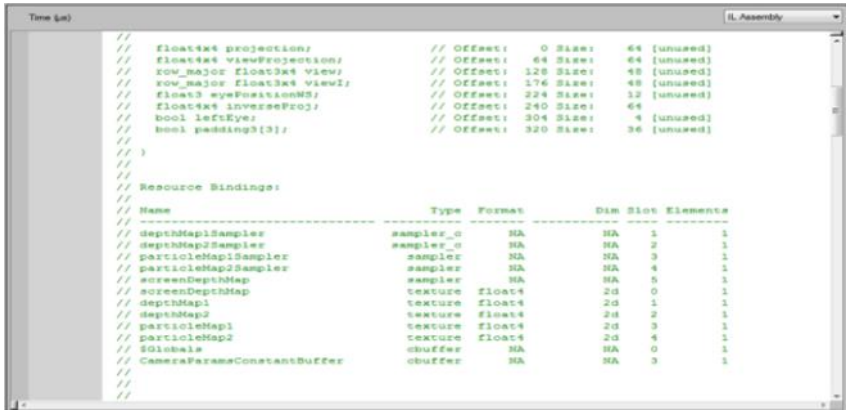
Fullscreen Shadow Pass

- Blocked on pixel shaders
  - Eu pixel shader stall = 42.3%
  - Originally read in 4 shadow textures
  - One shader for all quality settings
  - Clear textures read on lower settings



- Stencil mask added to remove selected areas, such as the sky

- Different shader used on medium and below, removed reads from particle shadow textures



Ripping Up The Rule Book

And you thought 15watts was a challenge...

A different approach to optimization

- Can remove high end PC features
  - Tablet GPU performance was initially ~53ms per frame
  - But more scope for making aggressive changes
  - Scalability (Lots more graphics menu options!)
  - More selective use of Specular and Normal Maps, etc
- Cheaper Shaders
  - Idea: use Environment Map shaders for our main scene?
  - This render pass is essentially a low quality version of our main colour pass
  - Too low quality in some cases
  - Saved 20ms GPU time!

Fixing the Visuals

- Using environment map shaders for main scene has consequences
  - Screen space maps all disappear (shadows, SSAO)
  - Seeing shader pass is a useful debug tool
    - Hard to see bugs in these shaders when looking directly into car reflection
- Fallback Render Pass
  - Engine already supports a fallback render pass
  - If Primary pass doesn't exist for shader, use Secondary
  - Implemented new pass to fix specific problems
    - Undercar shadow was missing
    - Headlights no longer illuminated the track at night
    - Etc

A bit more optimization

- Other ideas
  - Texture LOD bias
    - Visual quality declines very quickly, and tests currently show negligible gains
  - Lower geometry LODs
  - Nearer draw distances
  - Billboard LODs for trees/crowd
    - Reduces vertex cost, and lighting costs
  - Simple Post Process
    - Only need tone mapping (which, for us, requires bloom)
    - Motion blur, Lens flare, etc all gone

Low resolution particle rendering

- An effective optimisation for console/PC
  - Reduce fill-rate by rendering particles at lower resolution, and combining them with the main framebuffer
  - ¼ width and height
- Fixed-cost overheads a lot higher on Tablet
  - Creating downsampled depth buffer
  - Upsample downsampled colour buffer
- More efficient to render particles at full resolution, and sacrifice particle counts
  - High particle counts only occur during collisions, or when going off track

Too optimised?

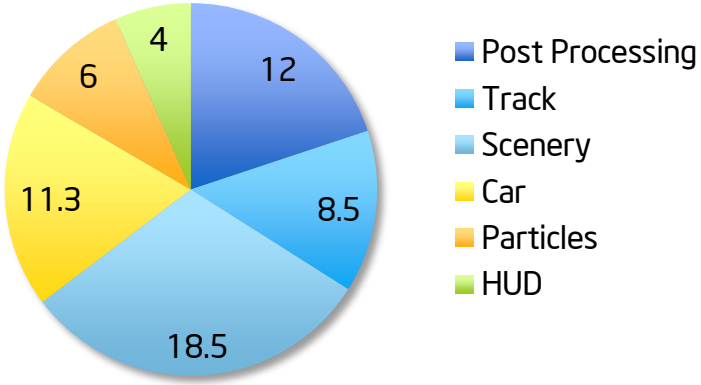
- Game now running at over 40fps!
- Lowest Preset used to be a “compatibility mode”
  - Particles, Crowd, Drivers, and Shadows were all disabled
  - Choose which systems we could turn back on to get us to 30fps

Upscaling is not just for consoles

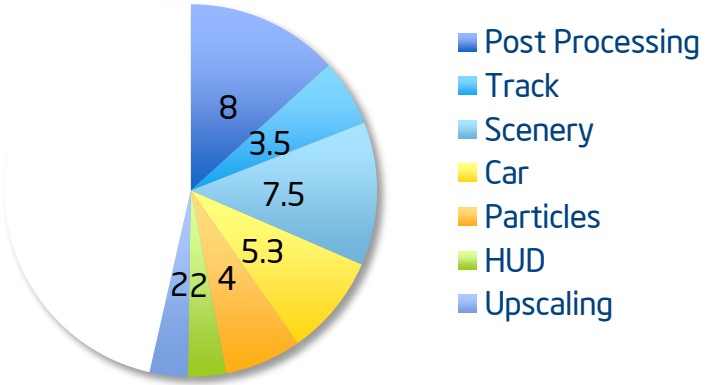
- HUD looked low quality at non-native Tablet resolutions
- High resolution backbuffer, low resolution colour render target
  - Already support naïve supersampling in engine
  - Easy to modify this feature to support smaller target, instead of larger

Performance Before and After

Pre Optimization



Post Optimization



That's a wrap:

1. New extensions allow for visual differentiation, are power efficient
2. Existing algorithms can be significantly optimized
3. Normal GPU optimization rules are subtlety different, bandwidth and power mean things are not always intuitive
4. CMAA good low cost Post-Process Anti-Aliasing solution on all hardware, for cases when concerned about blurriness / image degradation. Not as good AA results compared to SMAA though (especially vs. more expensive variations)
5. Sample & code available online at Intel web page
6. Are you making the best visual trade offs on power constrained hardware?

Can your engine scale from Tablet to high end PC????

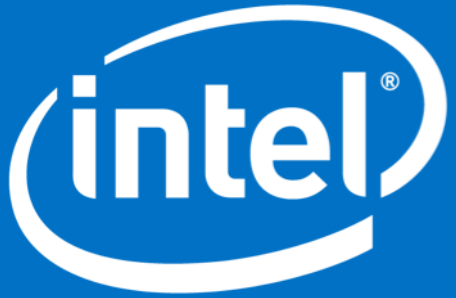
Acknowledgements

Robin Bradley, Peter Clark, Marco Alamia: Codemasters

Marco Salvi, Aaron Lefohn: Intel Advanced Rendering Technologies

Filip Strugar: CMAA and SSAO filip.strugar@intel.com

Questions?



Thanks for attending!
Want to go further?

Grid 2 http://tinyurl.com/buygrid2

AVSM research

OIT research

OIT research

AVSM sample

OIT sample

CMAA sample

Ready for More? Look Inside™.

Keep in touch with us at GDC and beyond:

- Game Developer Conference
  Visit our Intel® booth #1016 in Moscone South
- Intel University Games Showcase
  Marriott Marquis Salon 7, Thursday 5:30pm
  RSVP at **bit.ly/intelgame**
- Intel Developer Forum, San Francisco
  September 9-11, 2014
  intel.com/idf14
- Intel Software Adrenaline
  @inteladrenaline
- Intel Developer Zone
  software.intel.com
  @intelsoftware