SIGGRAPH2011 VANCOUVER

Filtering Approaches for Real-Time Anti-Aliasing

http://www.iryoku.com/aacourse/

Filtering Approaches for Real-Time Anti-Aliasing

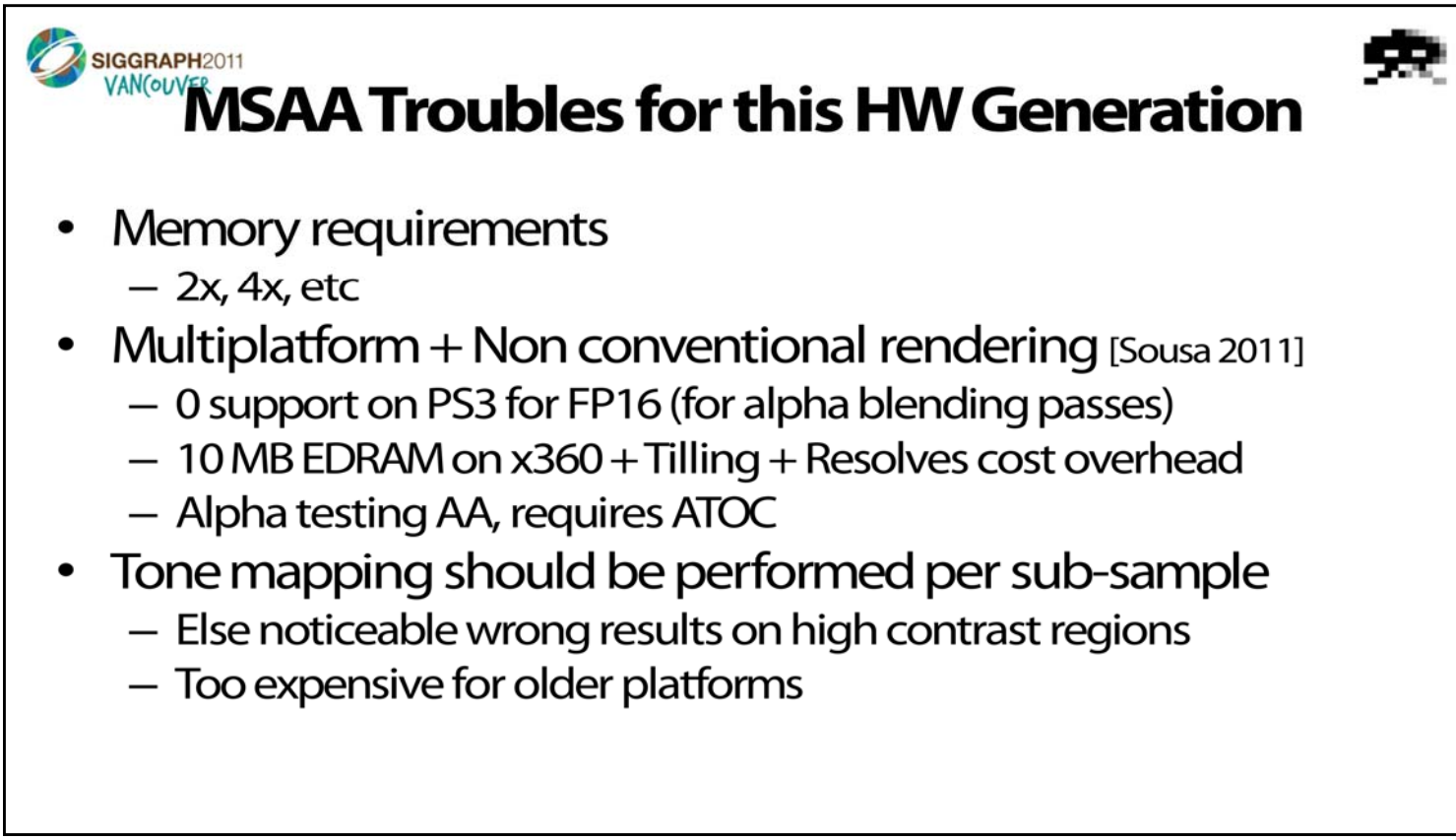
Anti-Aliasing Methods in CryENGINE 3

Tiago Sousa
R&D Principal Graphics Engineer
Crytek

tiago@crytek.com

CryENGINE 3 AA Requirements

- Orthogonal and general solutions
  - No per-platform AA solution
- Play nice with HDR/Deferred techniques
- Sub-pixel accuracy is important for us
  - Schimering was the biggest offender on Crysis 1 and 2 levels
  - Crysis had imensively aliased assets: alpha tested/tiny sub-pixel details
  - HDR makes it even worse, big range of lighting contrast/color variation
- Low memory footprint
- Cost less than 2 ms on low end GPUs
  - Every ms counts for consoles



Multitude of workarounds and added complexity

Each platform with its own solution, workarounds and optimizations

Performance cost

For current Gen, situation doens't look promising...

Minimal G-Buffer

- RGBA8 World Space BF Normals + Glossiness
- Readback Z24S8 Depth + Stencil for tagging interior areas

Deferred lighting buffers

- Xbox 360: 2x RGB10F resolved to RGB10A2 for performance/memory
- PS3: 2x RGBA8 encoded in RGBK
- PC: 2x FP16

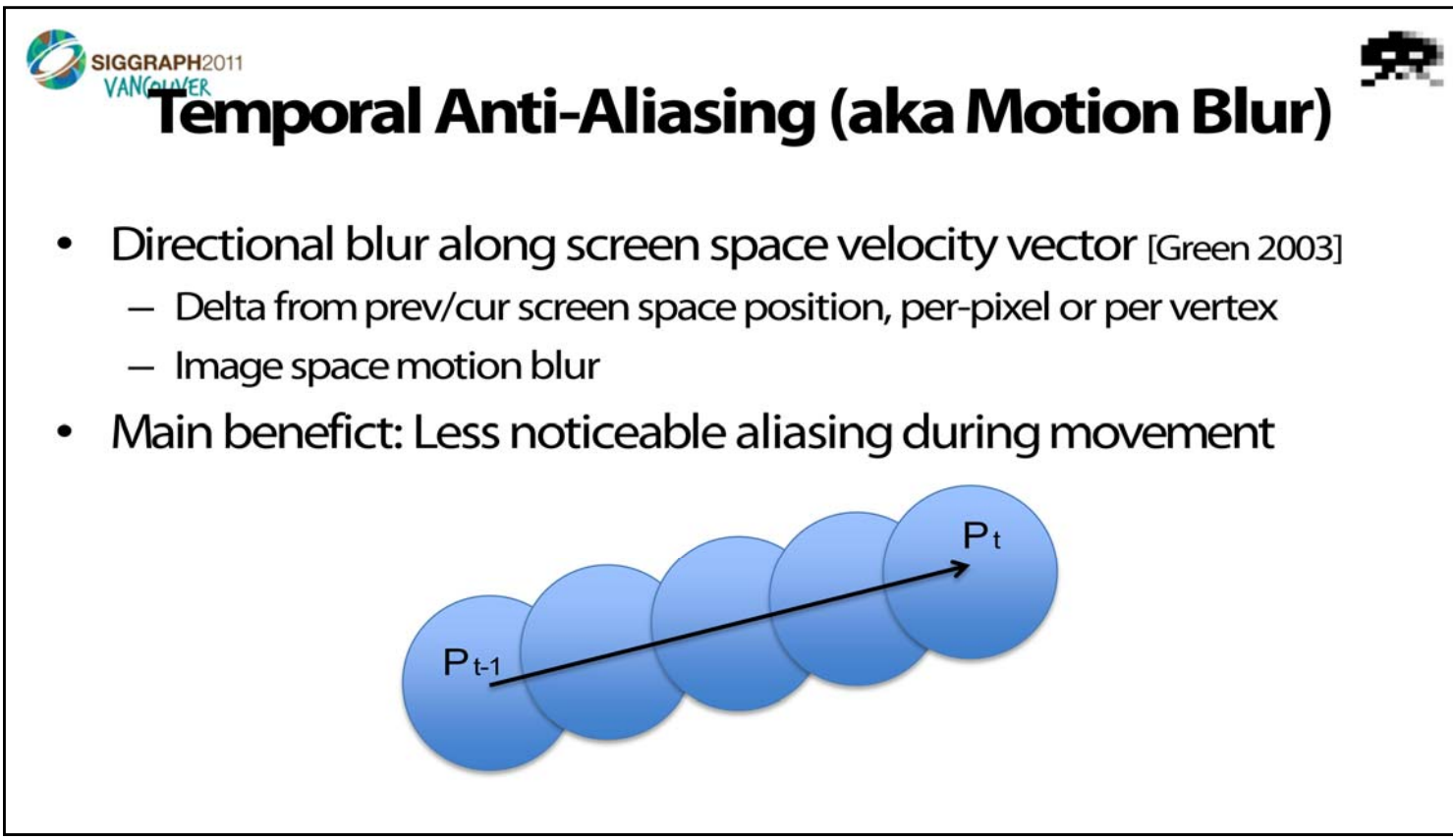
Scene buffer

- X360: RGB10F resolved to FP16 for performance
- PS3: RGBA8 encoded in RGBK for opaque, FP16 for transparents
- PC: FP16

X360: 28 MB; PS3: 31.5MB; PC: 49 MB at 720p, 110 MB at 1080p and so on.

SIGGRAPH2011 VANCOUVER

The Quest for AA Alternatives



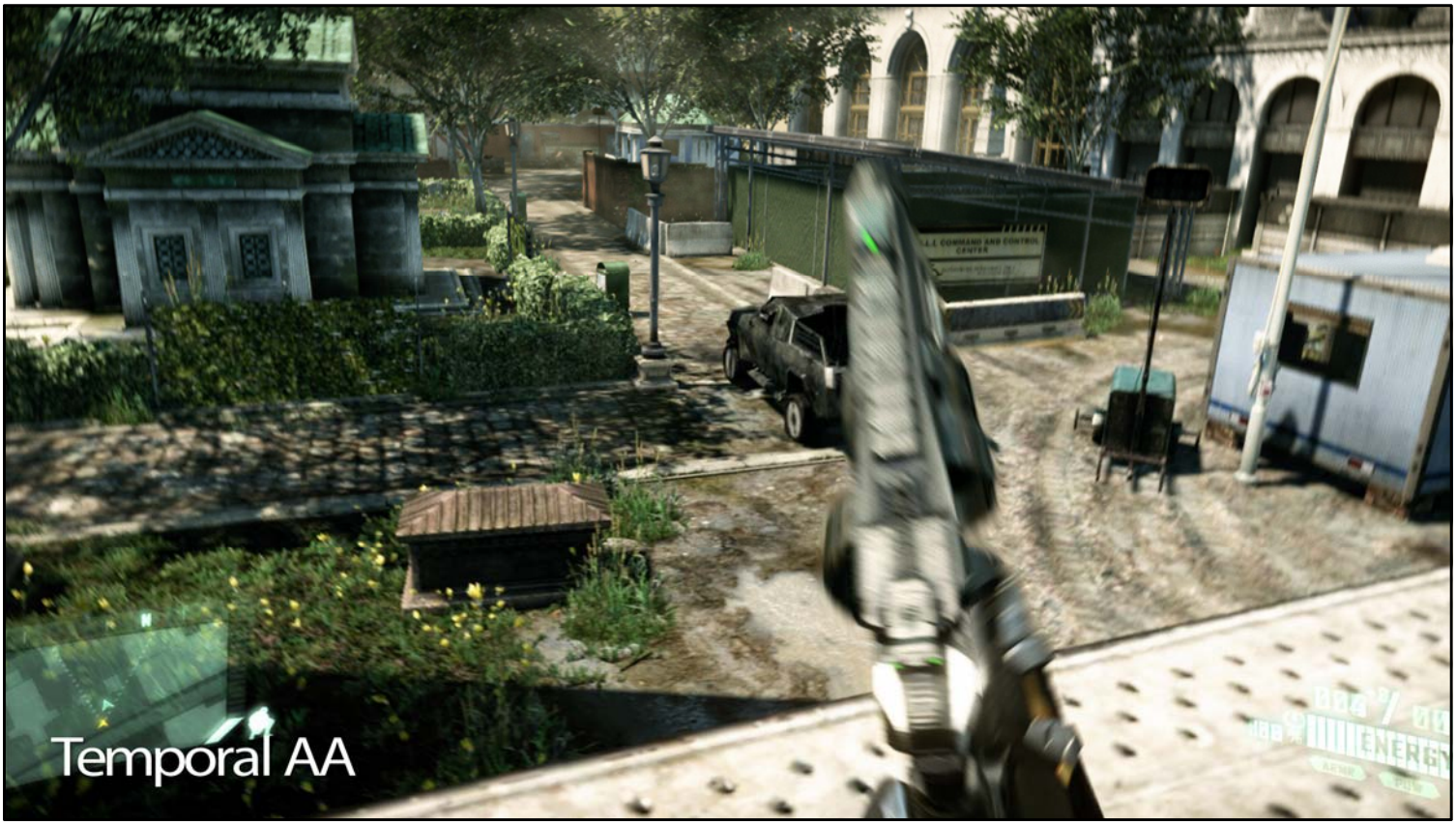
Increasingly popularity since Crysis 1 and KZ2

Aproximated as directional blur along screen space velocity vector

Delta from prev/cur screen space position, per-pixel or per vertex

Added bonus: Great cinematic look

User for all platforms


Temporal AA

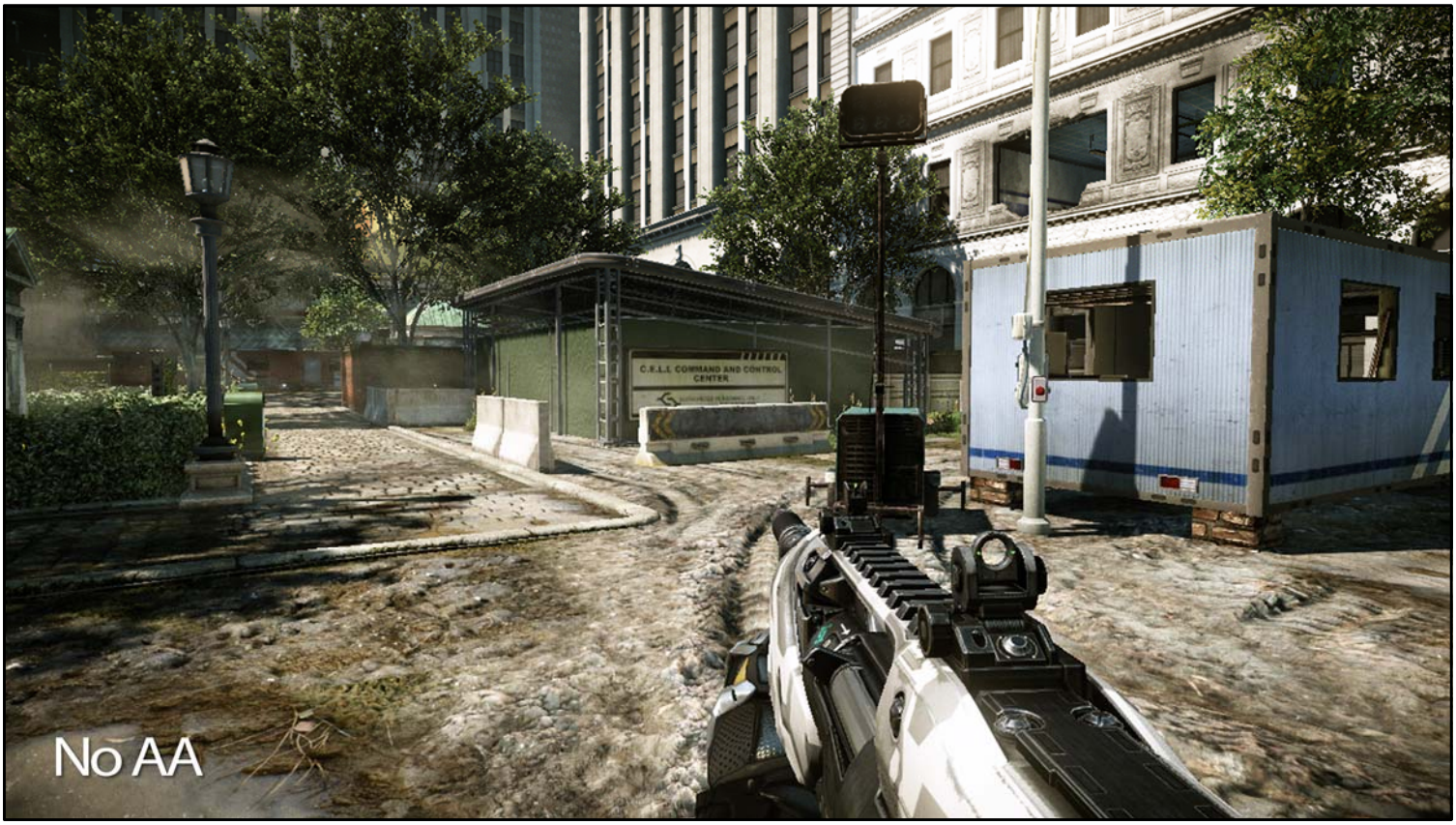
SIGGRAPH2011 VANCOUVER

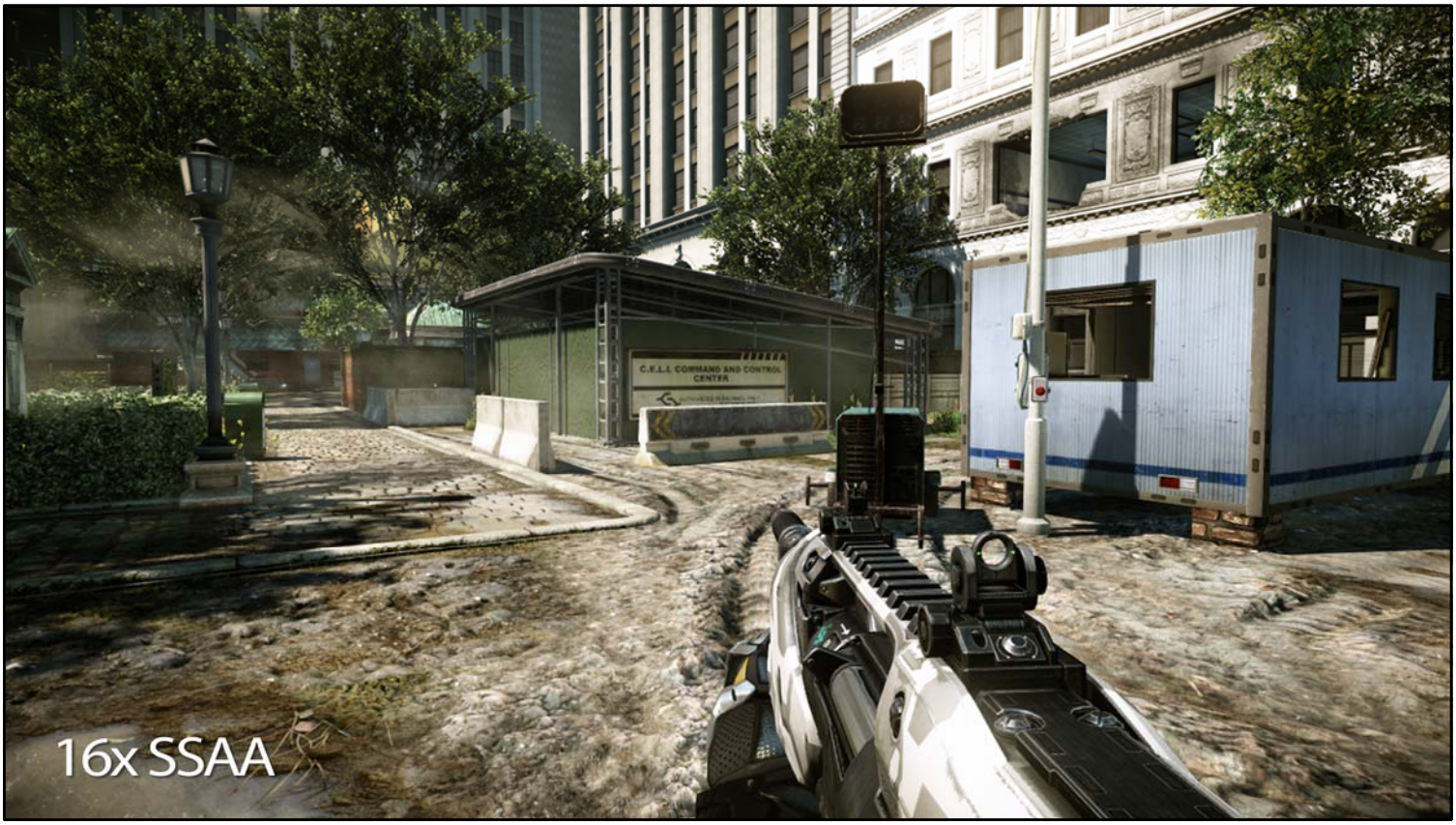
A-Buffer SSAA [Haeberli90]

- Add sub-pixel jitter to camera frustum
- Brute force: Render scene multiple times
  - N sub samples ⇔ N scene renders
- Robust and best quality
  - Also more uses besides SSAA (TSSAA/DOF/Soft-Shadows)
- Base concept used for our techniques

- Problem: Cannot afford render scene multiple times (yet)
  - Great for reference/marketing quality shots though

Alternatively, the usual approach of rendering higher resolution and downscale

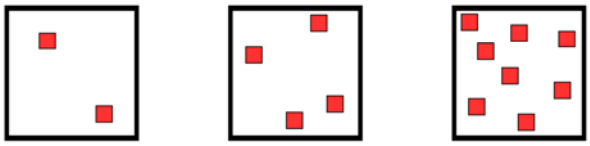

C.R.I.S. COMMAND AND CONTROL CENTER
No AA


C.R.L.L COMMAND AND CONTROL CENTER
16x SSAA

SIGGRAPH2011 VANCOUVER

Distribute A-Buffer SSAA Overframes

- Running at 60 fps ?
  - Add sub-pixel jitter to camera frustum every frame
  - Store previous/current frame and linear blend them
  - Light-speed 2x SSAA: ~0.5 ms on current consoles
  - 2 frames ⇔ 2x SSAA, 4 frames ⇔ 4x SSAA, etc



- But... not many reach 60 fps on consoles
  - Lower fps results in extremelly noticeable image ghosting

Used in few older 60 fps games (DMC4?) , but who was first?

And even at 60 fps, still noticeable ghosting

Strafing is the worst case, we'll use if for images

Minimizing Artifacts

- Improving blending: Reprojection
  - Velocity vector fetches from previous frame sub-sample target
  - Exactly same as in TAA (but single tap)
- Deformable geometry slightly more expensive to handle
  - Output pixel velocity into a render target
  - Could not afford for vegetation
- Problem: Disocluded regions ghosting


PULL
Using reprojection

Minimizing Artifacts (2)

- Disable blending if ||V|| > 0?
  - Very rare the case when player not moving
  - And we still want AA during camera movement
- Weighting using color/edge tagging ?
  - Sub-pixel/hi frequency detail results in noticeable schimering
- Reprojection range clamping
  - Pixel weight proportional to reprojection limit
    - Eg: fBlendW = saturate(1 - (fVLen / fVMaxLen))
  - Coarse Depth stored in sub-sample buffer alpha channel
    - Mask out if fVLen > fMaxVThreshold and fCurrD > fPrevD

Disable accumulation if ||V|| > 0?

Very rare the case when player not moving

And we still want AA during camera movement

Weighting using color ?

Sub-pixel/hi frequency detail results in noticeable schimering

Reprojection range clamping ? (used for Crysis 2)

Pixel weight proportional to reprojection limit

Eg: fBlendW = saturate(1 - (fVLen / fVMaxLen))

Coarse Depth stored in sub-sample buffer alpha channel

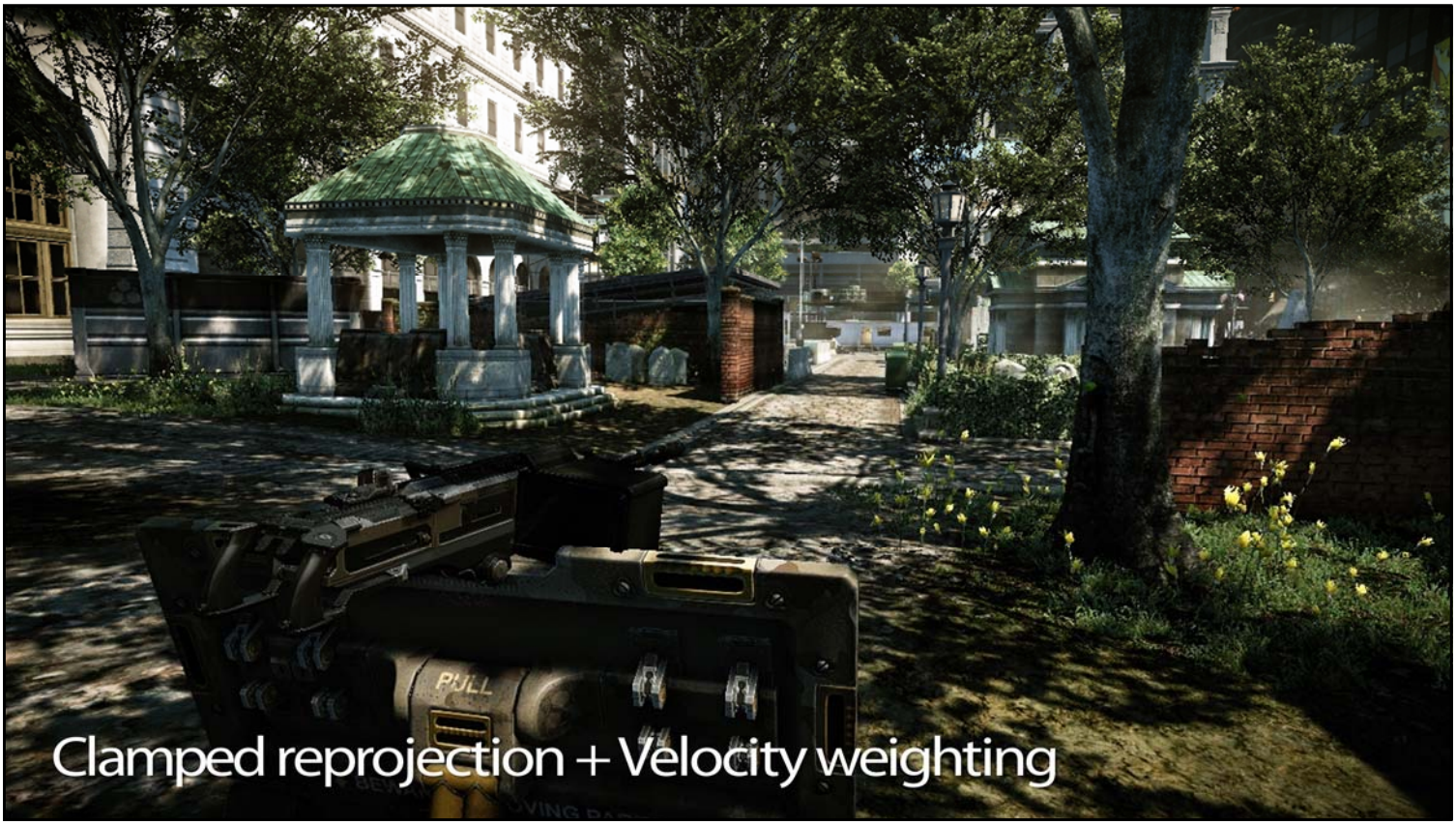
Mask out if fVLen > fMaxVThreshold and fCurrD > fPrevD

8 bit insufficient to cover large range, limited to nearby


Clamped reprojection (used in Crysis2)

Minimizing Artifacts (3)

- Store ||V|| in sub-sample buffer alpha channel
  - Weight: abs(fPrevLenV – fCurrLenV) / fVMaxLen


Clamped reprojection + Velocity weighting

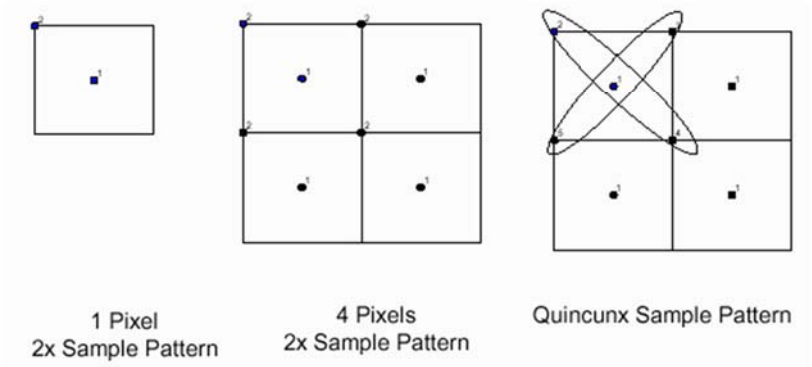
SIGGRAPH2011 VANCOUVER

Example Code

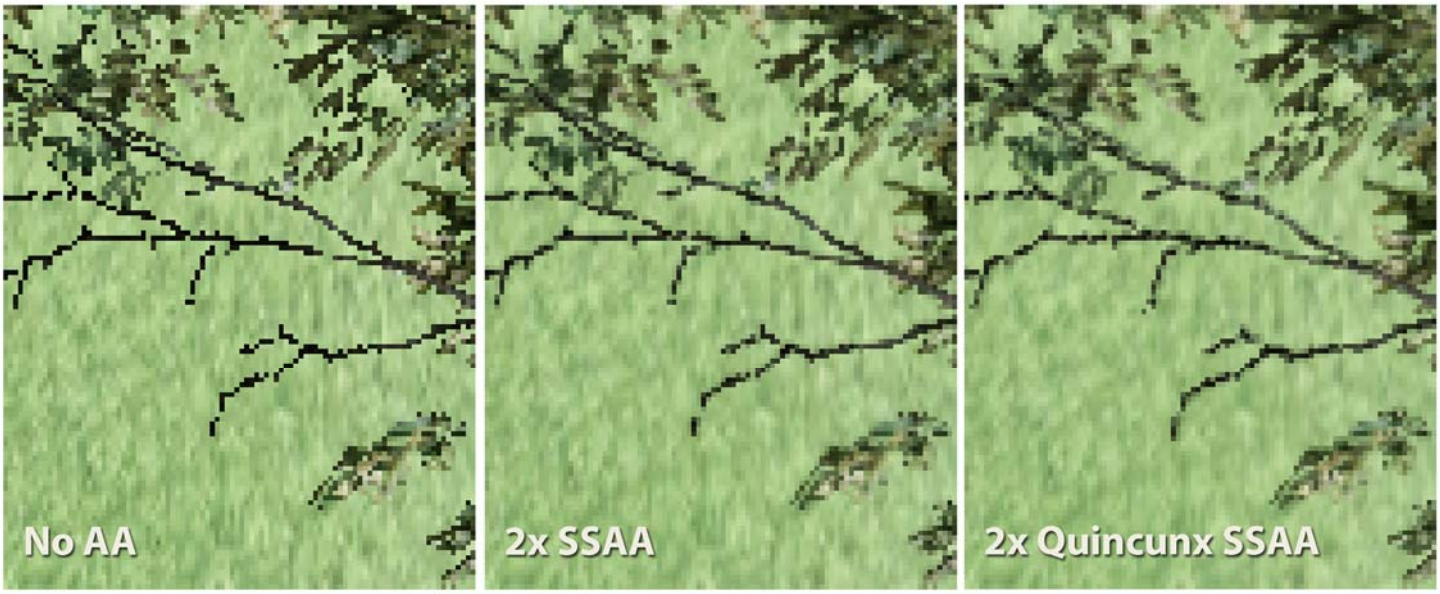
```
float fDepth = GetLinearDepth(sDepth, tcBase.xy );
float3 vPosWS = WorldViewPos.xyz + IN.vCam.xyz * fDepth;

float4 vPrevPos = mul(mViewProjPrev, float4(vPosWS, 1.0 ));
vPrevPos /= vPrevPos.w;

float2 vVelocity = vPrevPos.xy - tcBase.xy;
half4 cObjVelocityParams = tex2D(sObjVelocity, tcBase.xy) ;
half2 vObjVelocity = DecodeMotionVector( cObjVelocityParams );

vVelocity = cObjVelocityParams.w? vObjVelocity : vVelocity;
float fVLenSq = dot(vVelocity.xy, vVelocity.xy) + 1e-6f;
vVelocity /= fVLenSq;

half4 cCurr = tex2D(sCurrFrame, tcBase.xy)
half4 cPrev = tex2D(sPrevFrame, tcBase.xy + vVelocity * min(fVLenSq, fVMaxLen) );

half fBlendW = 0.5 - 0.5 * saturate(fVLenSq / fVMaxLen );
fBlendW *= saturate(1 - (abs(cCurr.a - cPrev.a) * fVWeightScale  );

OUT.Color = lerp(cCurr, cPrev, fBlendW);
```

2x Quincunx SSAA

- Improving quality with 2 sub-samples
  - Bilinear fetch to one of sub-samples
  - "Aproximate" 4x SSAA



2x Quincunx SSAA



Distributed A-Buffer SSAA: Caveats

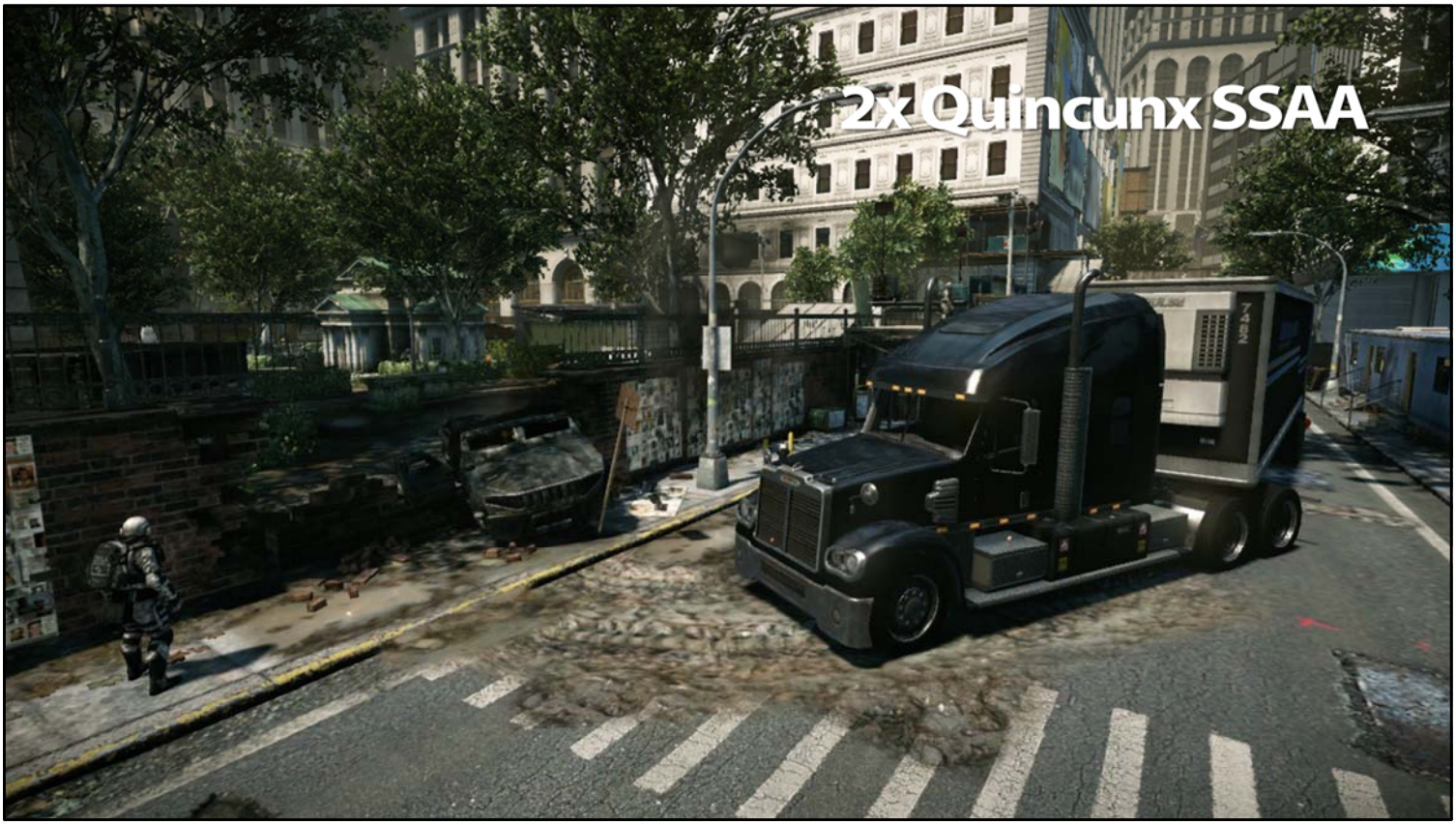
- Not temporally stable
  - No AA on disocluded regions
  - Input signal changes (color/lighting), no robust solution yet
- Alpha blending problematic
  - Withouth OIT, only possible to handle correctly for first hit
  - Additional overhead
- Multi-GPU
  - Additional frame latency to address
  - For Crysis 2, we switched to Nvidia's FXAA when in MGPU
    - Schimering was again, biggest complain from MGPU users

Future Work

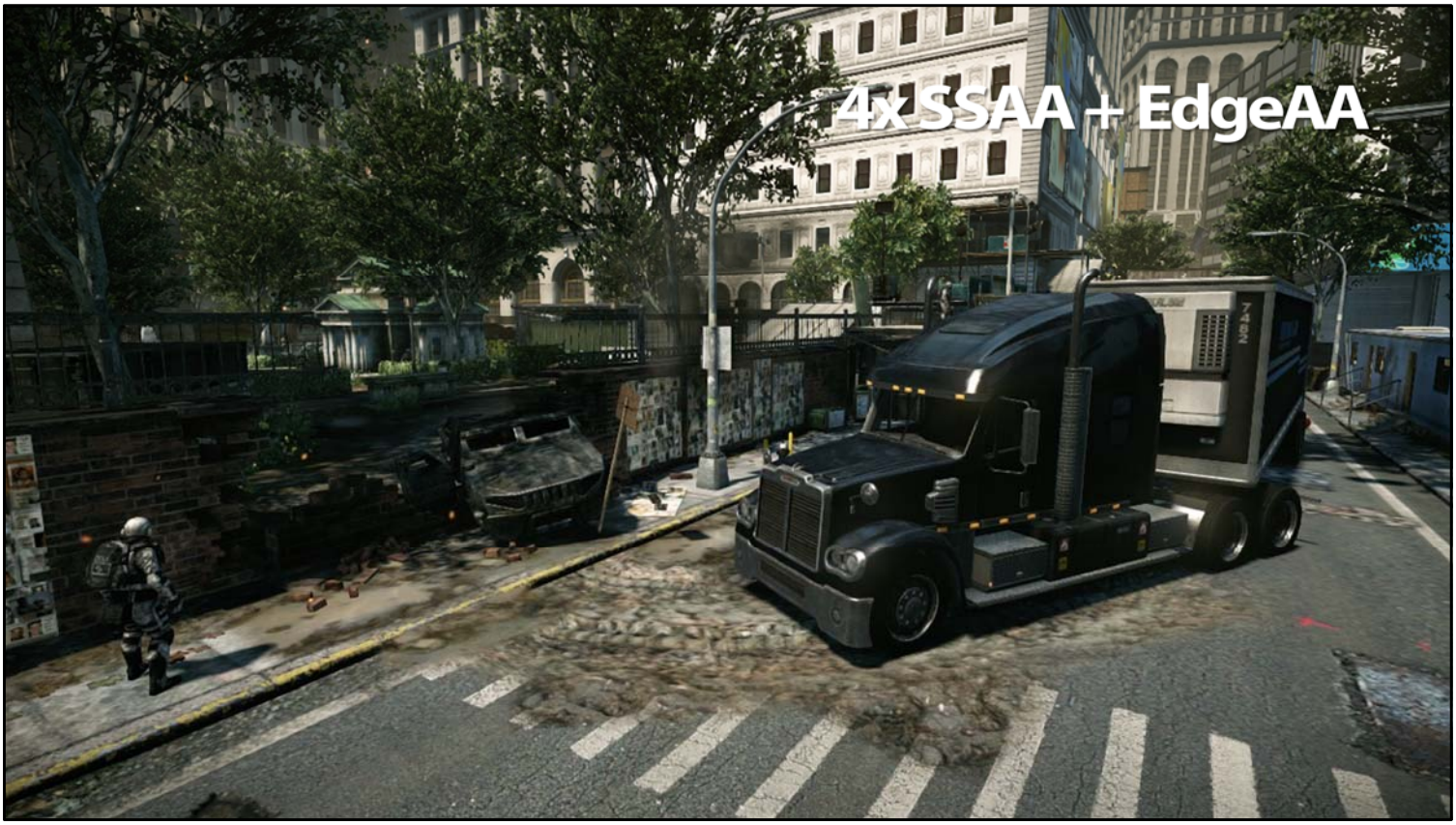
- SSAA combo with post processed AA
  - Maybe similarly to DLAA: horizontal/vertical edges, blend taps
    - This means at least 4 additional taps
  - AA on disocluded regions


No AA


2x SSAA


2x Quincunx SSAA


4x SSAA

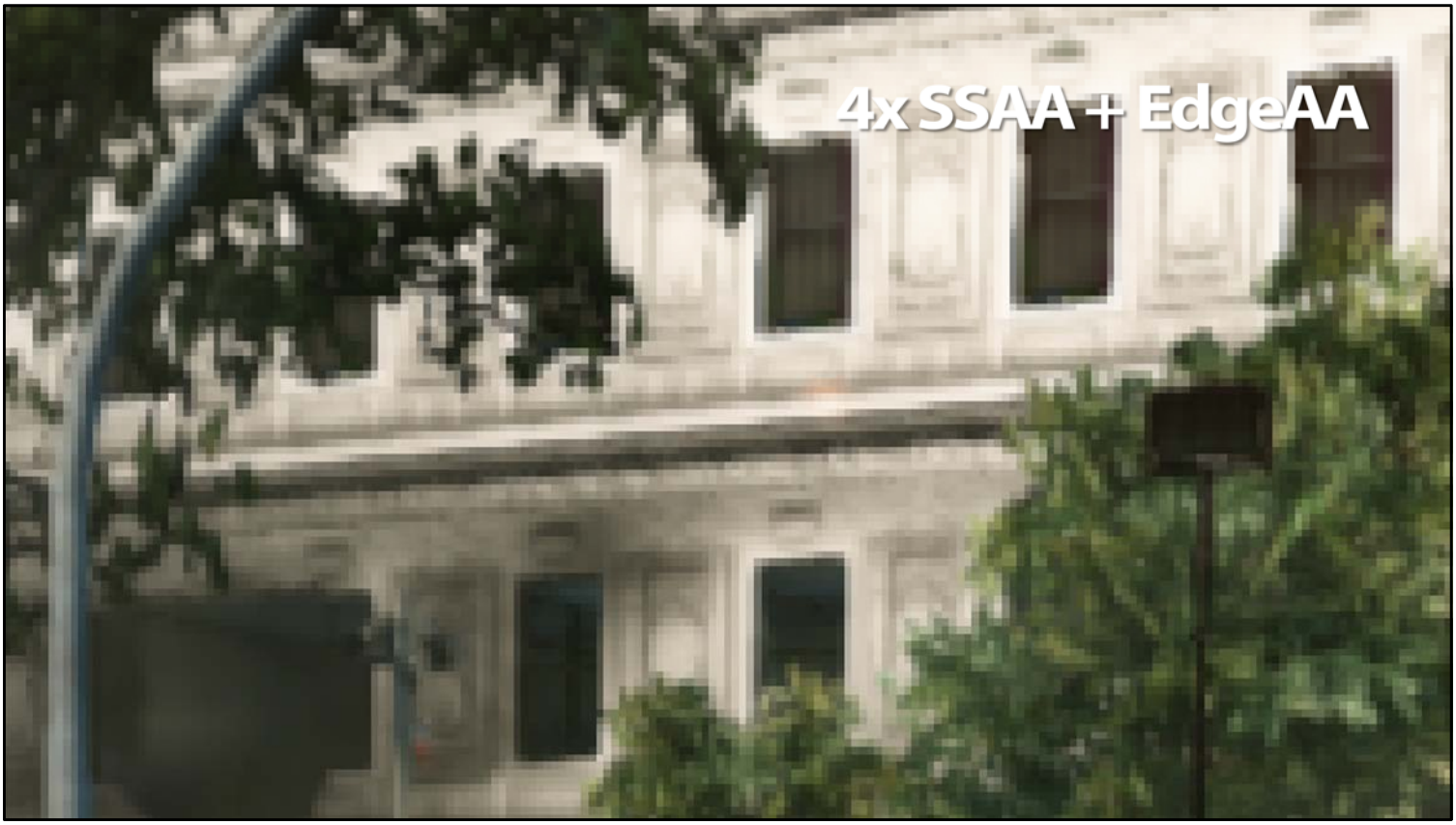

4x SSAA + EdgeAA

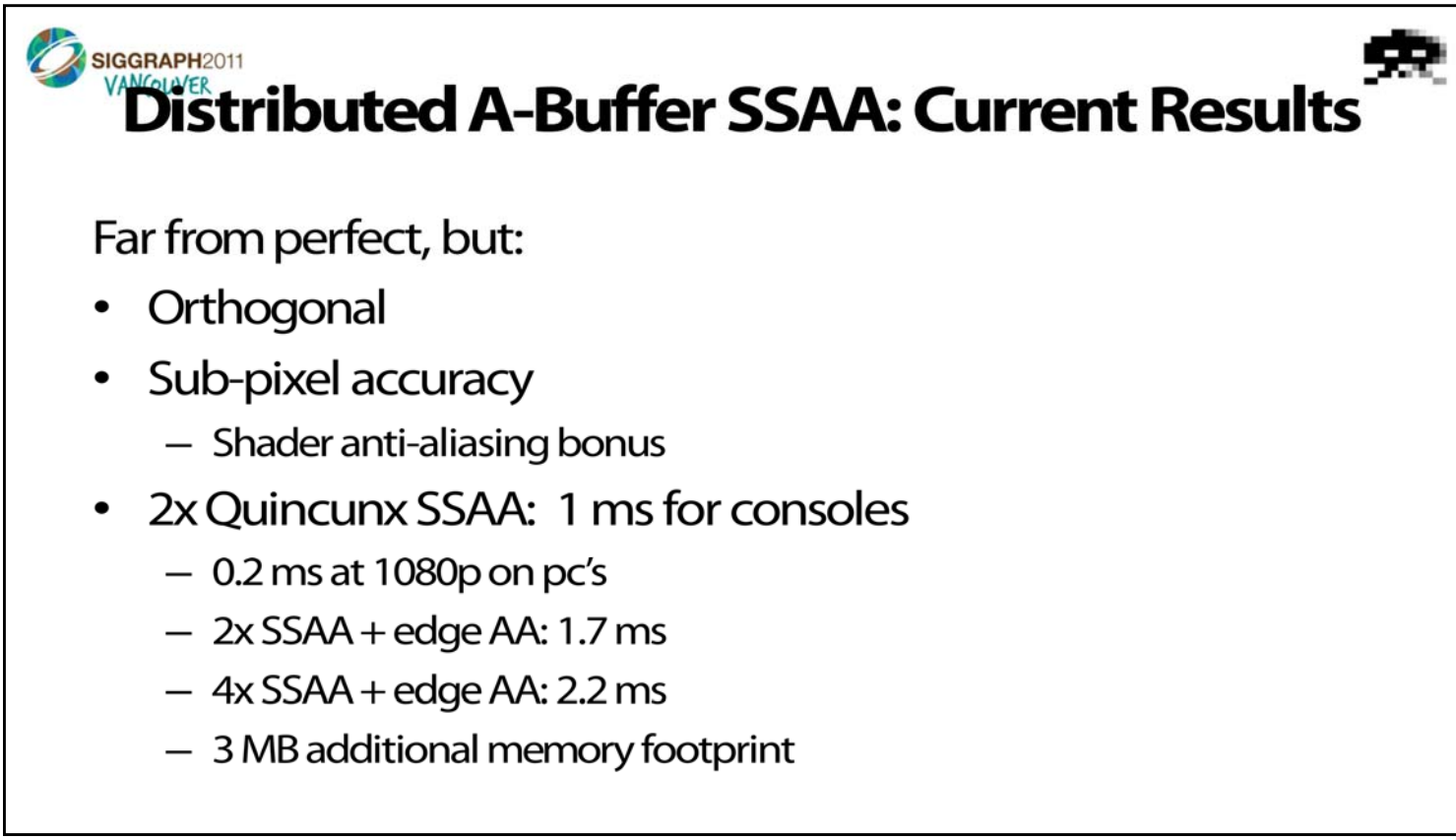

No AA


2x SSAA


2x Quincunx SSAA


4x SSAA


4x SSAA + EdgeAA



Missing pretty pictures

SIGGRAPH2011 VANCOUVER

Acknowledgements

- Nick Kasyan,Nicolas Schulz, Vaclav Kyba, Michael Kopietz, Carsten Wenzel, Vladimir Kajalin, Andrey Konich, Ivo Zoltan Frey
- Jorge Jimenez, Diego Guitierrez, Naty Hoffman

- And to the entire Crytek team

Further Readings

Haeberli, P, Akeley, K "The Accumulation Buffer: Hardware Support for High-Quality Rendering", 1990

Siggraph'96 Course , Blythe, D et al "Programming with OpenGL: Advanced Rendering", 1996

Green, S "Stupid OpenGL Shader Tricks", 2003

Sousa, T. "Crysis Next Gen Effects", 2008

Swoboda, M "Deferred Rendering in FrameRanger", 2009

Yang, G et al "Amortized Super Sampling", 2010

Binks, D. "Dynamic Resolution Rendering", 2011

Sousa, T., Kasyan, N. and Schulz, N. "Secrets of the CryENGINE 3 Technology", 2011

SIGGRAPH2011 VANCOUVER

Questions ?

tiago@crytek.com

twitter: crytek_tiago

SIGGRAPH2011 VANCOUVER

Bonus: Marketing Screenshots

If time allows

Bonus: Marketing Screenshots

- Always some trickery
  - On CryENGINE 2 rendered multiple tiles at big resolutions and downsampled to get SSAA
- On CryENGINE 3 distributed SSAA with many samples
  - Random sub-pixel jitter
  - Almost perfect SSAA
  - All Crysis 2 marketing shots used this variation