The Evolution of Variable Rate Shading In Games

Marissa du Bois

Senior Graphics Engineer

03/2023

VRS IN GAMES & ENGINES

VRS In Games & Engines

VALAR in World of Warcraft

Integrating XeSS & VALAR

Improving Visual Quality

VRS Tier 1 & 2 On Intel® GPUs

Variable Rate Shading Tier 1

Intel® 10th Gen Processor Graphics (code named Ice Lake)	Introduced in 2019	Set Shading Rates Per Draw Call
1x1, 1x2, 2x1, 2x2 2x4, 4x2, and 4x4	DirectX 12 API: RSSetShadingRate	

iGFX to dGFX

intel ARC GRAPHICS

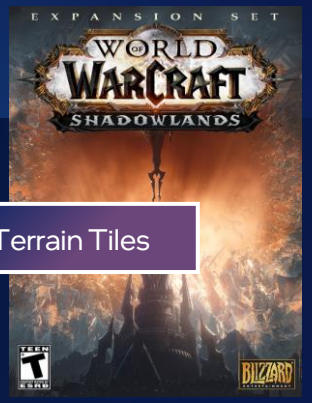
Variable Rate Shading Tier 2

Intel® Arc Graphics A-series (code named Alchemist)	Introduced in 2022	Intel's First Mainstream Discrete GPU
Image Based & Per-Prim VRS	DirectX 12 API: RSSetShadingRateImage	

intel ARC

Titles Enabled With VRS Tier 1 & Tier 2

VRS Tier 1



Terrain Tiles

Blizzard Entertainment®
2020

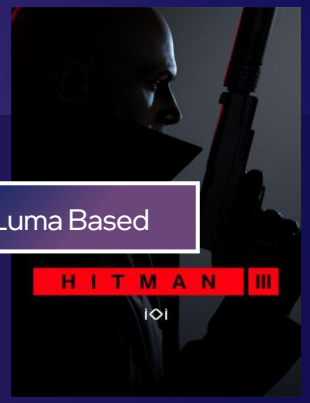
Pre-Pass

CAPCOM®
2021

Particles

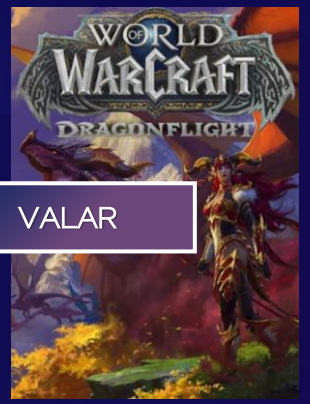
GIANTS Software
2022

VRS Tier 2



Luma Based

IO Interactive
2021



VALAR

Blizzard Entertainment®
2022

VRS Tier 1 – Per-Material

Relaxed Mode Plus

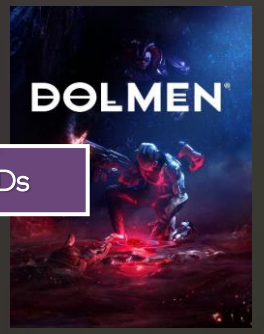
Per-Pass

Per-Material

Tripwire Interactive
2021

UNREAL ENGINE

VRS LODs



Massive Work Studio
2022

intel ARC

VRS Content Published By Intel

Resource	URL	Author	Publisher	Published
Getting Started With VRS Tier 1	https://www.intel.com/content/www/us/en/developer/articles/guide/getting-started-with-variable-rate-shading-on-intel-processor-graphics.html	Adam Lake, Laura Reznikov & Marissa du Bois	Intel	2019
Using Variable Rate Shading to improve the User Experience	https://software.intel.com/en-us/videos/use-variable-rate-shading-vrs-to-improve-the-user-experience-in-real-time-game-engines	Adam Lake, Filip Strugar, Kelly Gawne, Trapper Mcferron	Intel	2019
VRS Tier 1 With Chivalry 2 (GDC 2020)	https://www.youtube.com/watch?v=d-qEvmVcg8I	Marissa du Bois & John Gibson	Intel	2020
VRS Tier 1 Usage Guide With Unreal Engine 4	https://www.intel.com/content/www/us/en/developer/articles/guide/variable-rate-shading-tier-1-usage-guide.html	Marissa du Bois & Jeff Rous	Intel	2020
Velocity And Luminance Adaptive Rasterization (VRS Tier 2)	https://www.intel.com/content/www/us/en/developer/articles/technical/velocity-luminance-adaptive-rasterization-vrs-tier.html	Marissa du Bois, Adam Lake, Meghan Weicht, Aria Kraft	Intel	2022
Velocity And Luminance Adaptive Rasterization (GDC 2022)	https://www.youtube.com/watch?v=T48dW9nvuvI	Adam Lake, Marissa du Bois	Intel	2022

VRS TIER 2 WITH VALAR

VRS In Games & Engines

VRS Tier 2 With VALAR

VALAR in World of Warcraft

Integrating XeSS & VALAR

Improving Visual Quality

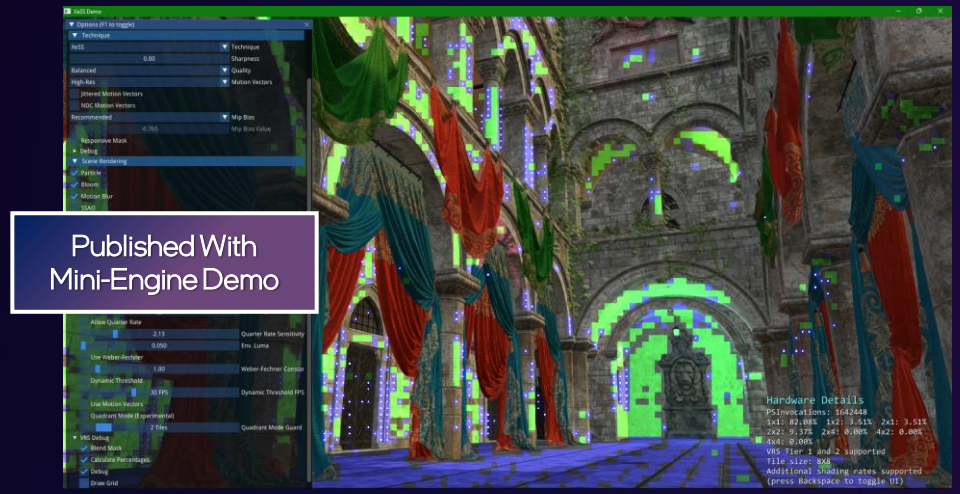
Velocity & Luminance Adaptive Rasterization

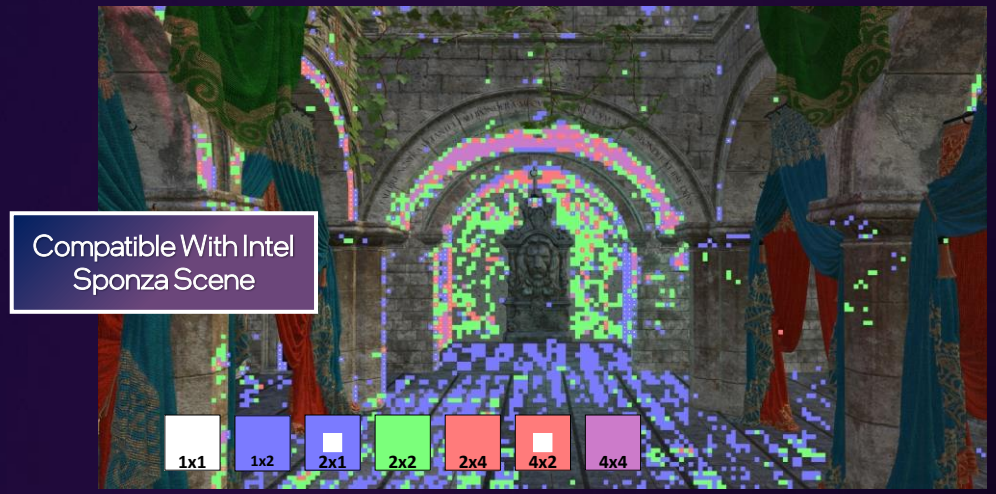
Background

- Whitepaper Published By Intel® in 2022
- Mini-Engine Demo Released on GitHub in 2022
- Based on Adaptive Shading by Lei Yang, 2019
- Approximates “Perceptually Lossless” Output
- Weber-Fechner Contrast Laws
- Luminance Mean-Squared-Error (MSE)
- Just-Noticeable-Difference (JND) Algorithm

Adjustable Input Parameters

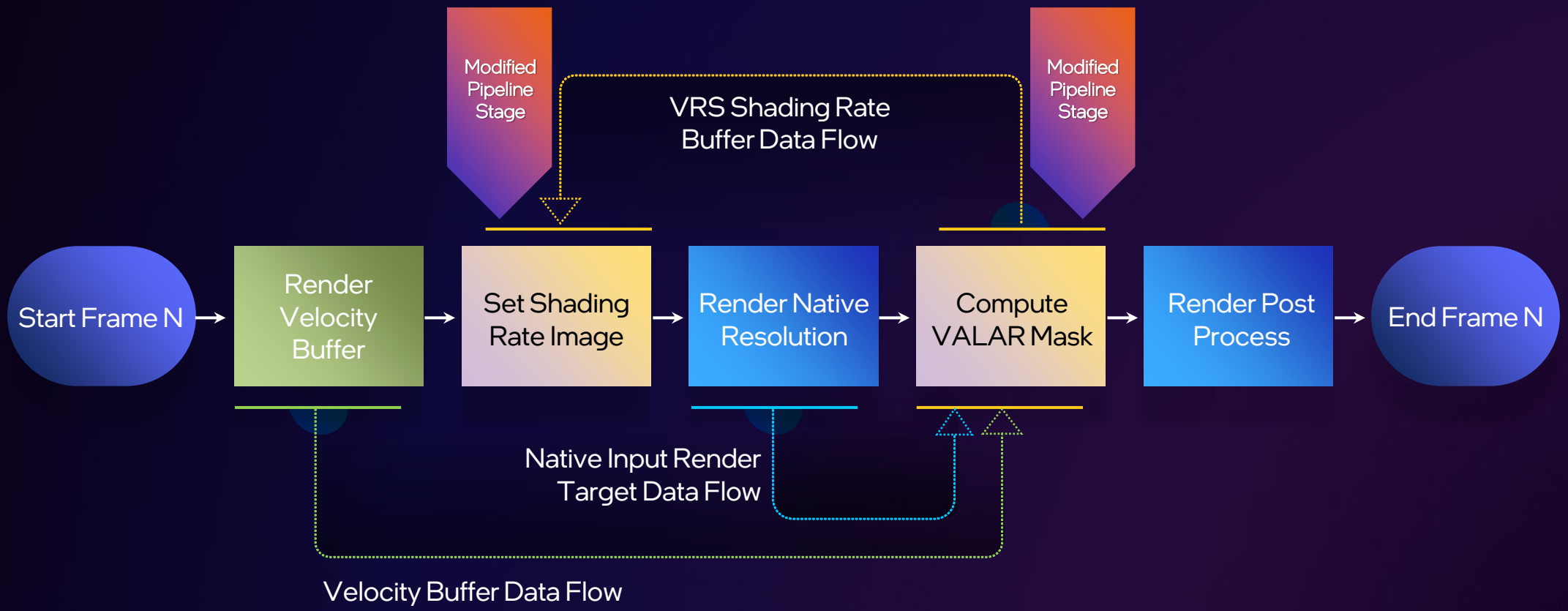
- Sensitivity Threshold (T)
- Environment Luminance (E)
- Quarter Rate Shading Modifier (K)
- Weber-Fechner Mode (C)
- Motion-Vector Mode





https://www.intel.com/content/www/us/en/developer/articles/technical/velocity-luminance-adaptive-rasterization-vrs-tier.html

VALAR Rendering Pipeline



Tile Based Luminance

Convert RGB To Luminance

$$L_p = R \cdot 0.212671 + G \cdot 0.715160 + B \cdot 0.072169$$

Sum The Tile Luminance

$$L_t = \sum L_p$$

Average The Tile Luminance

$$\bar{L}_t = \frac{L_t}{N}$$

Linear RGB Input

Screen Space Luma

Mean-Squared-Error

Sum X & Y Luma Differences Per Tile

$$\Delta_x = \sum \left| \frac{L_x - L_{x-1}}{2} \right|$$

$$\Delta_y = \sum \left| \frac{L_y - L_{y-1}}{2} \right|$$

Approximate Mode Use Weber-Fechner Mode For Tunable Precision

Average X & Y Luma Differences

$$\bar{\Delta}_x = \frac{\Delta_x}{N}$$

$$\bar{\Delta}_y = \frac{\Delta_y}{N}$$

Compute Mean Squared Error (MSE)

$$\varepsilon_x = \sqrt{\bar{\Delta}_x}$$

$$\varepsilon_y = \sqrt{\bar{\Delta}_y}$$

Used in the **Just-Noticeable-Difference** Evaluation

8x8 Luma Tiles

Average Tile Luma

Just-Noticeable-Difference

Compute The Just-Noticeable-Difference

$$J = T \cdot (\bar{L}_t + E)$$

T & E Configured by User

Branchless JND & MSE Evaluation

$$R_x = R_2 \cdot \neg\big((\varepsilon_x \geq J) \vee (\varepsilon_x < J)\big) + R_4 \cdot (\varepsilon_x < J)$$

$$R_y = R_2 \cdot \neg\big((\varepsilon_y \geq J) \vee (\varepsilon_y < J)\big) + R_4 \cdot (\varepsilon_y < J)$$

Branchless Logic Evaluates to 0 or 1

Terms Are Mutually Exclusive

Combine X & Y Shading Rates

$$R = R_x \ll 2 \mid R_y$$

X & Y Shading Rate

Final Shading Rate

intel ARC

Motion Vectors [Optional]

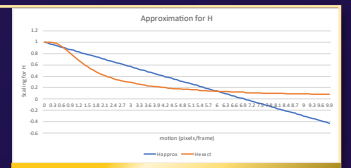
Find The Minimum Tile Velocity

$$V_t = min(Vp, Vt)$$

Half & Quarter Rate Velocity Modifiers [Yang, 2019]

$$V_h = \left(\frac{1}{1 + (1.05 \cdot V_t)^{3.10}}\right)^{0.35}$$

$$V_q = 2.13 \cdot \left(\frac{1}{1 + (0.55 \cdot V_t)^{2.14}}\right)^{0.49}$$



Substitute With Linear Approximation

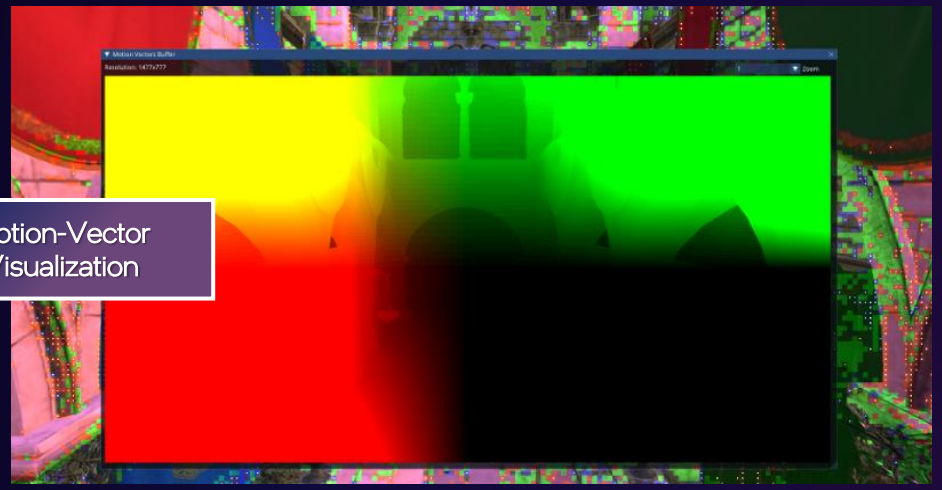
Combine X & Y Shading Rates

$$R_x = R_2 \cdot \neg\left((V_h \cdot \varepsilon_x \geq J) \vee (V_q \cdot \varepsilon_x < J)\right) + R_4 \cdot (V_q \cdot \varepsilon_x < J)$$

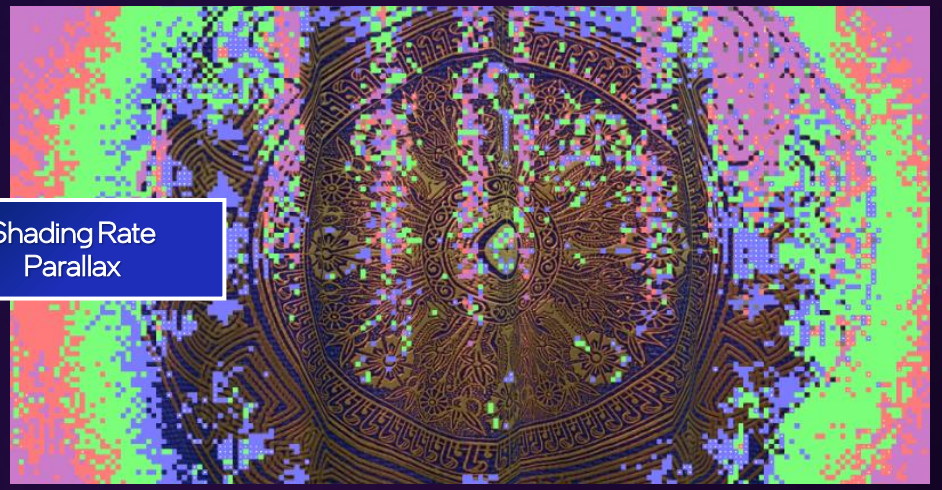
$$R_y = R_2 \cdot \neg\left((V_h \cdot \varepsilon_y \geq J) \vee (V_q \cdot \varepsilon_y < J)\right) + R_4 \cdot (V_q \cdot \varepsilon_y < J)$$

MSE Modified By Velocity

Precompute Repeated Expressions

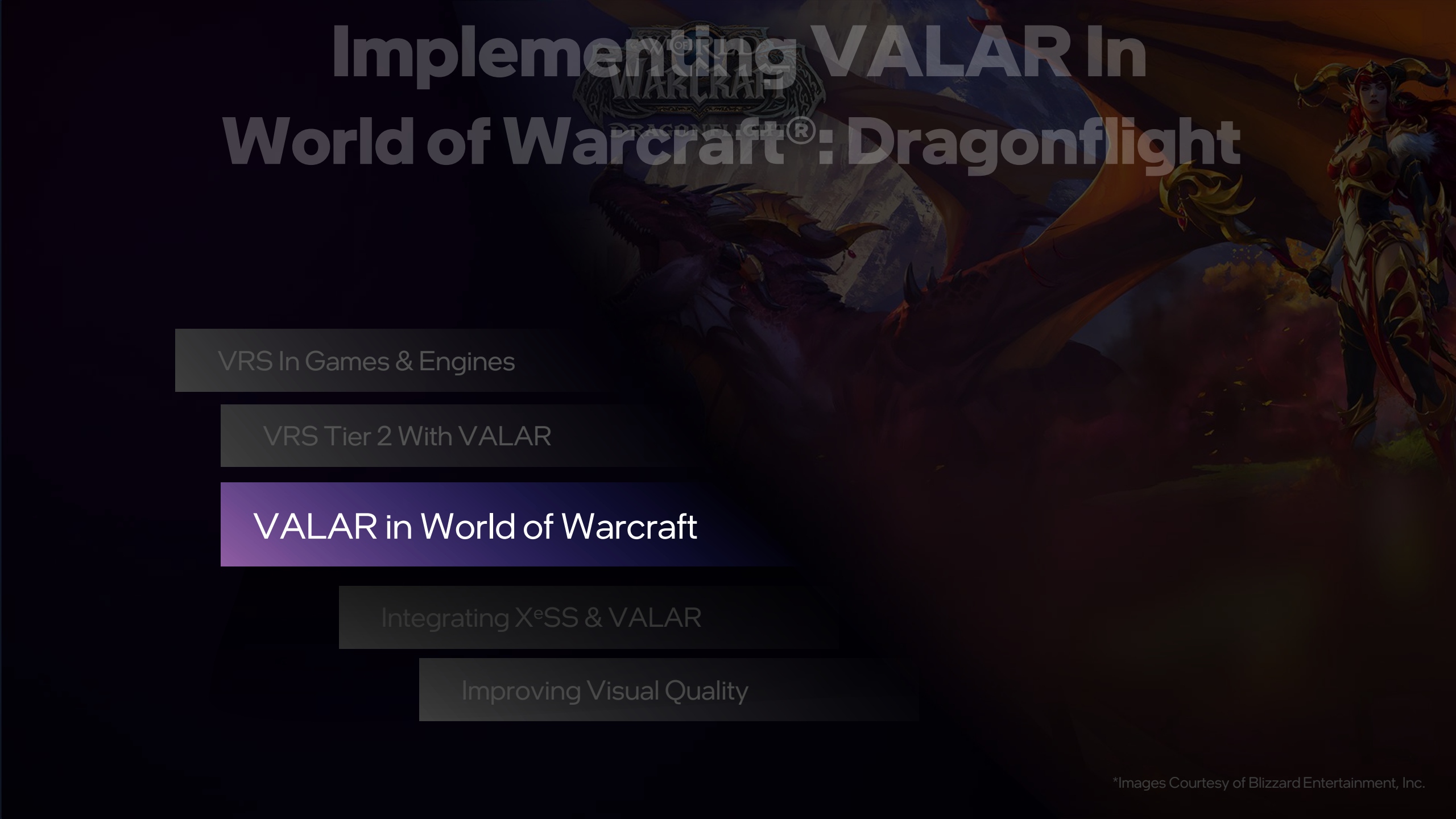


Motion-Vector Visualization



Shading Rate Parallax

Implementing VALAR In World of Warcraft®: Dragonflight



- VRS In Games & Engines
- VRS Tier 2 With VALAR
- **VALAR in World of Warcraft**
- Integrating XeSS & VALAR
- Improving Visual Quality

*Images Courtesy of Blizzard Entertainment, Inc.

VALAR In World of Warcraft

Background

- Disabled, Standard, and Aggressive Modes
- NPCs & Player Models Rendered at 1x1
- Opaque Geometry Uses Screen Space Shading
- Doodads Rendered With Screen Space Shading

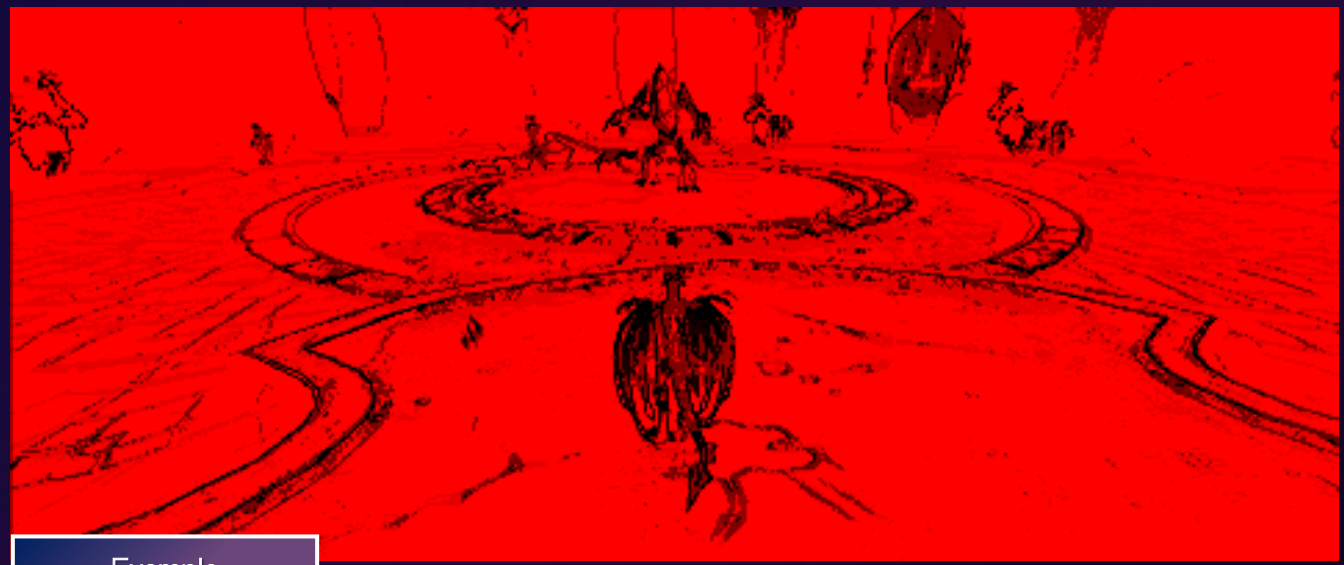
Default Configuration

- Standard Sensitivity: 0.31
- Aggressive Sensitivity: 0.50
- Quarter Rate Shading Modifier: 2.13
- Environment Luminance: 0.05
- Motion Vector Mode: Disabled
- Weber-Fechner Mode: Disabled

Considerations

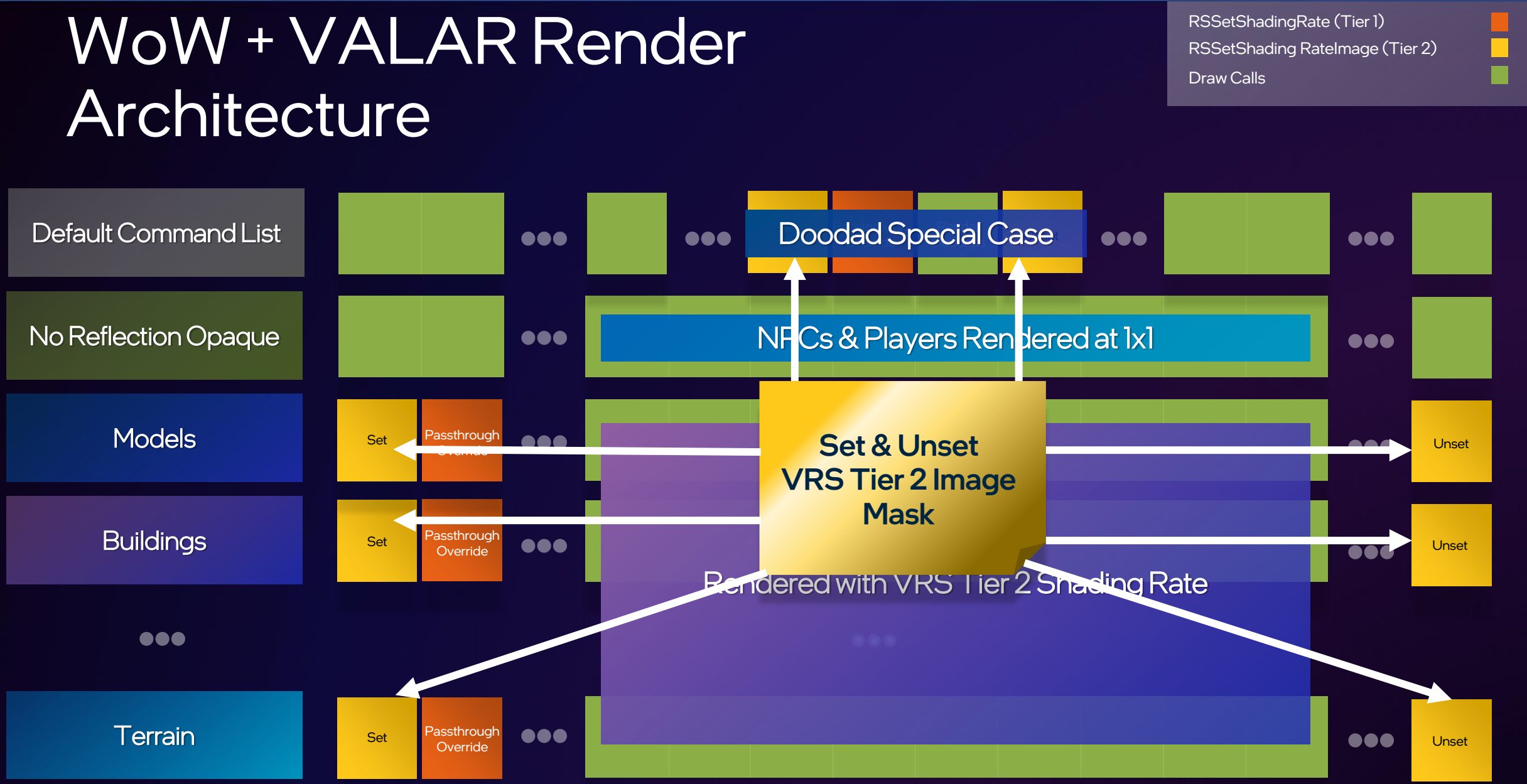
- Uses Multiple Command Lists
- Doesn't Support Motion Vectors
- Manual Alpha-Test Visual Corruption

VRS Tier 2 Settings

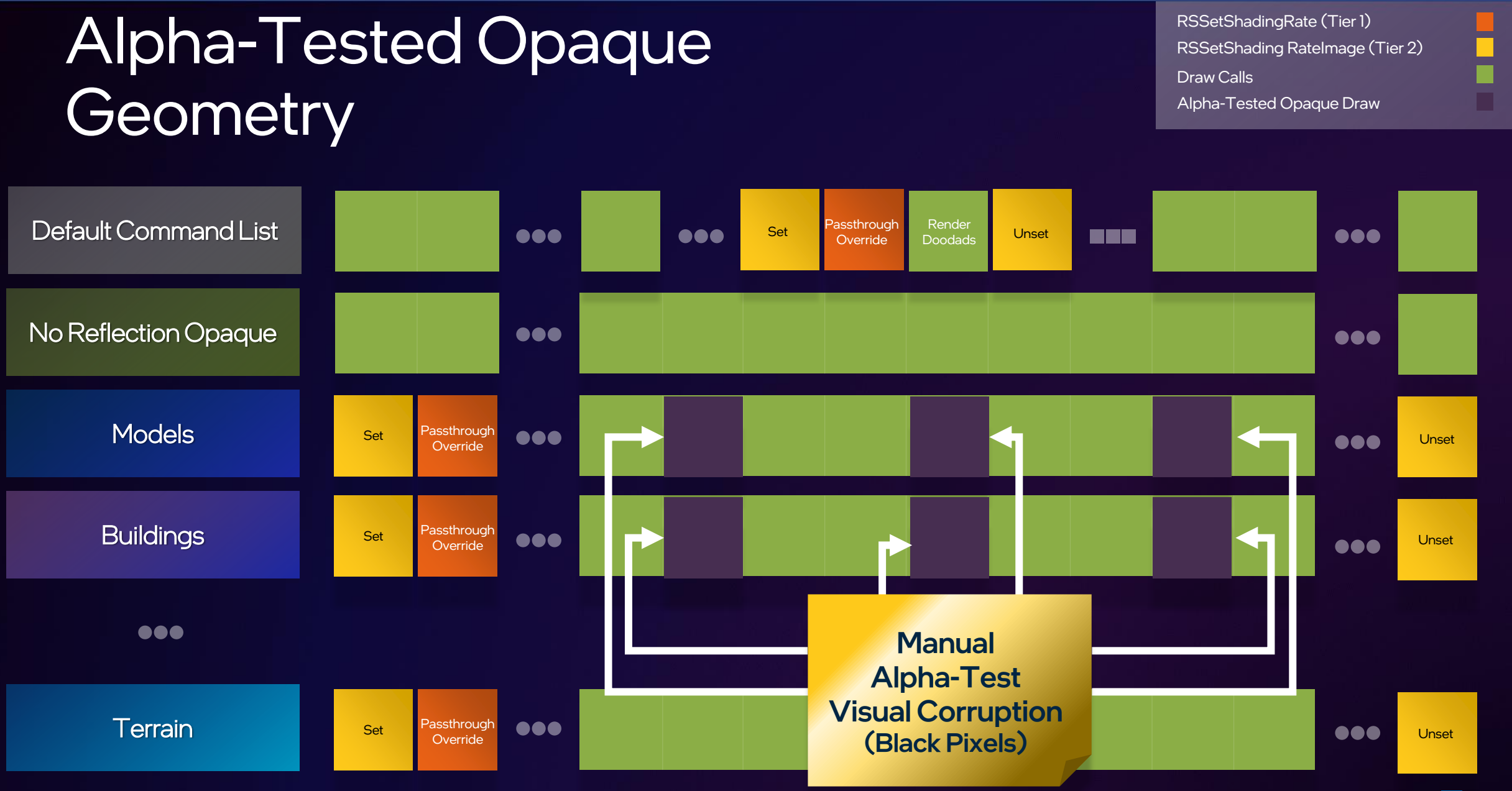


Example VRS Buffer

WoW + VALAR Render Architecture



Alpha-Tested Opaque Geometry

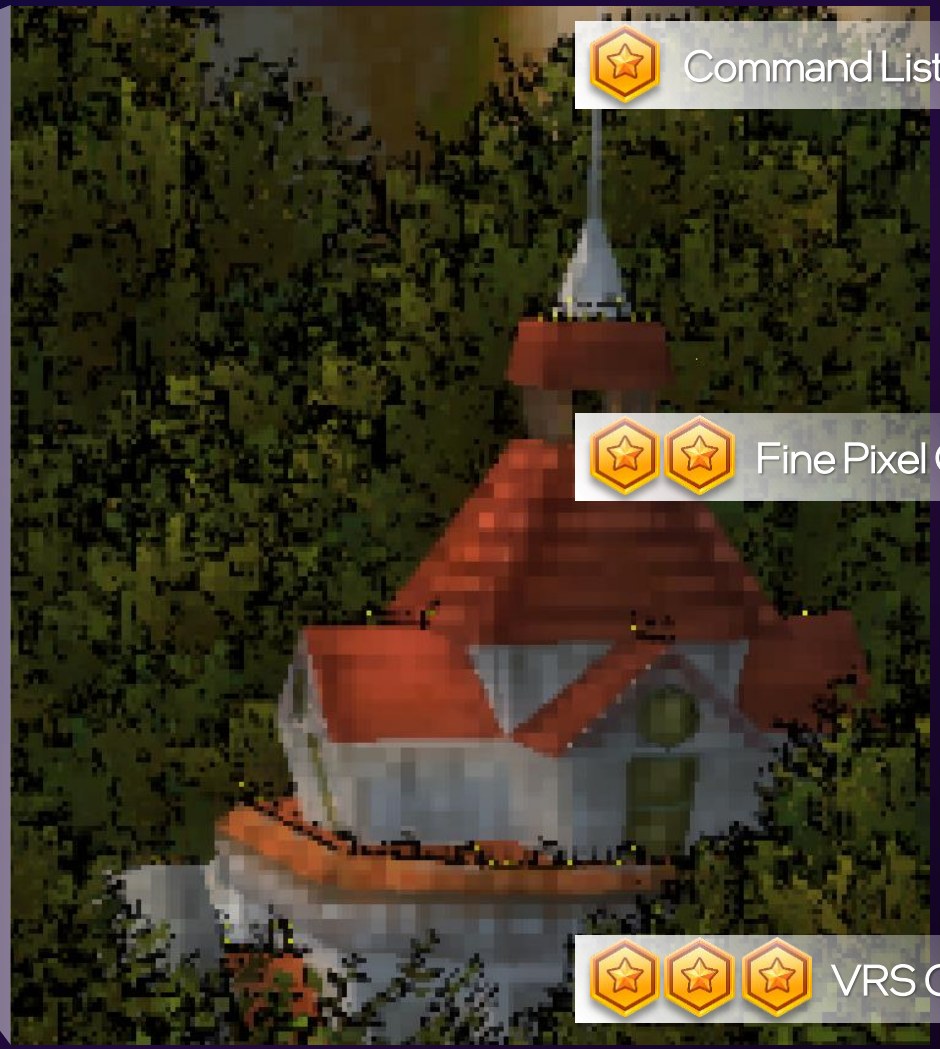


Manual Alpha-Test Visual Corruption

- Black Pixels
- Edges of Foliage
- Models & Buildings
- Manual Alpha-Test
- Uses Discard

*Images Courtesy of Blizzard Entertainment, Inc.

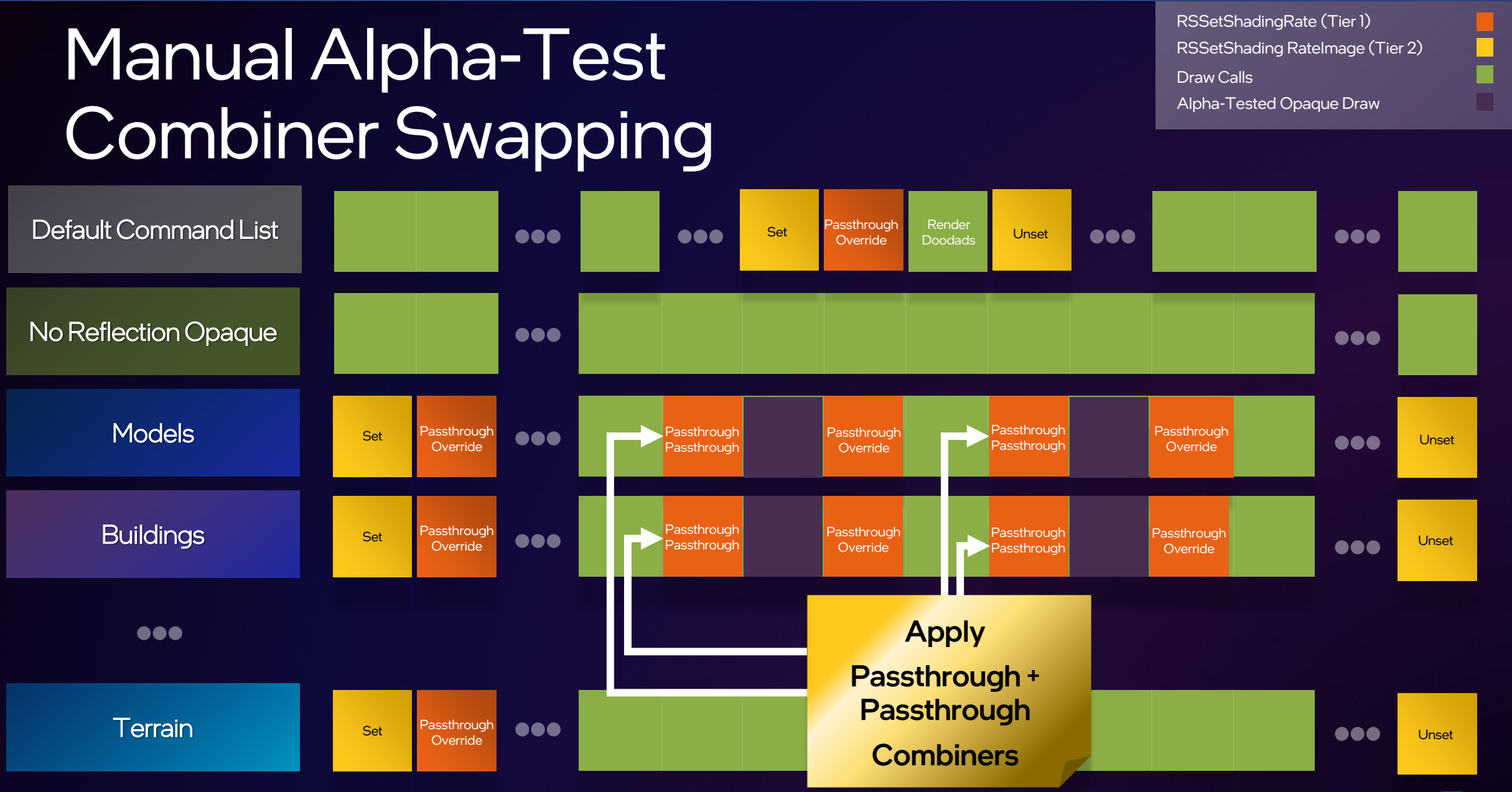
Manual Alpha Test –
Visual Corruption Example*



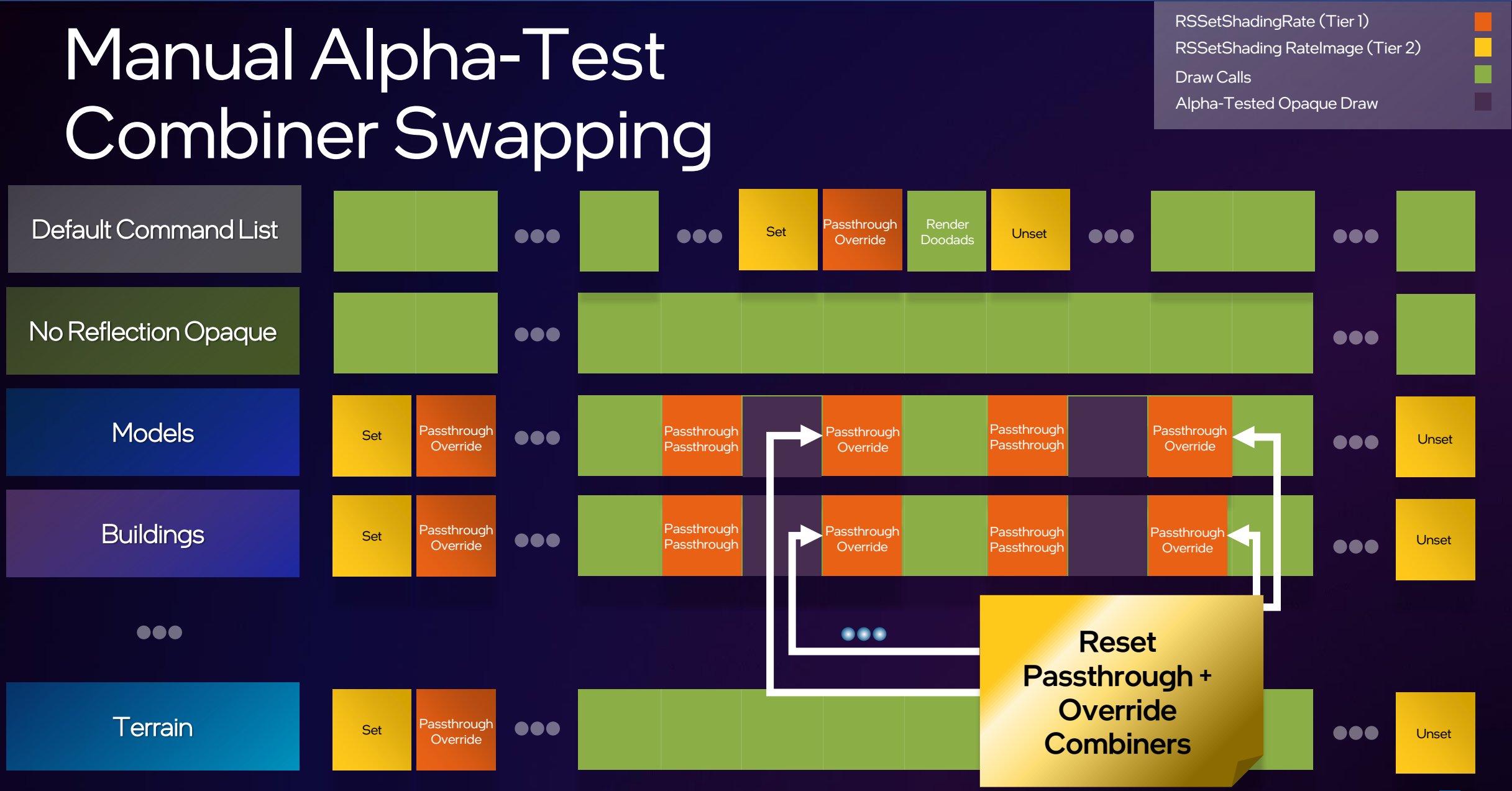
Manual Alpha Test – Cropped & Zoomed **400%***

intel ARC

Manual Alpha-Test Combiner Swapping



Manual Alpha-Test Combiner Swapping



Using VRS Combiners

Combiners Can Be Applied

- Per-Draw (PS)
- Per-Provoking Vertex (VS)
- Per-Primitive (GS)
- Image Based (CS/CPU)

Used in WoW

Used in WoW

Types of VRS Combiners

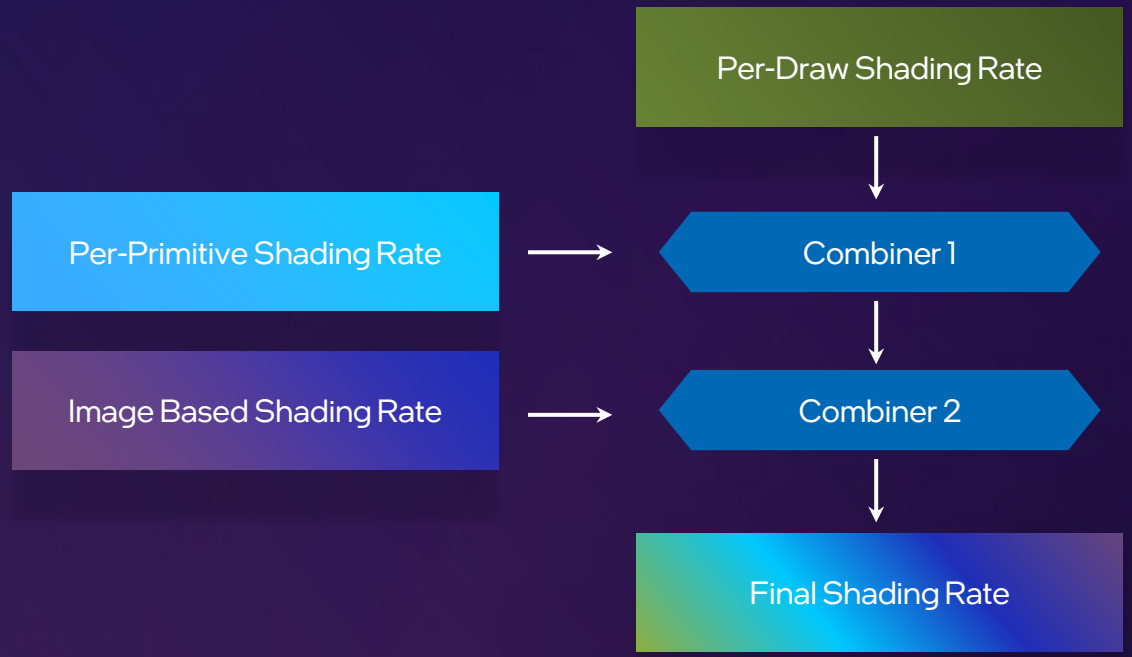
- Passthrough
- Override
- Min
- Max
- Sum

Check Hardware Features

DirectX 12 Combiner API

- void RSSetShadingRate(D3D12_SHADING_RATE baseShadingRate, const D3D12_SHADING_RATE_COMBINER *combiners);

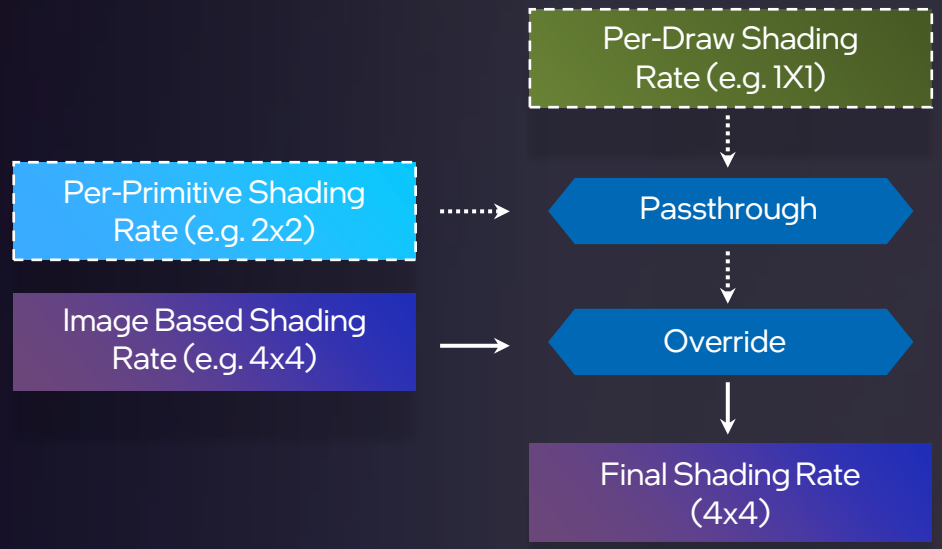
VRS Combiner Flow



intel ARC

VRS Combiners in World of Warcraft

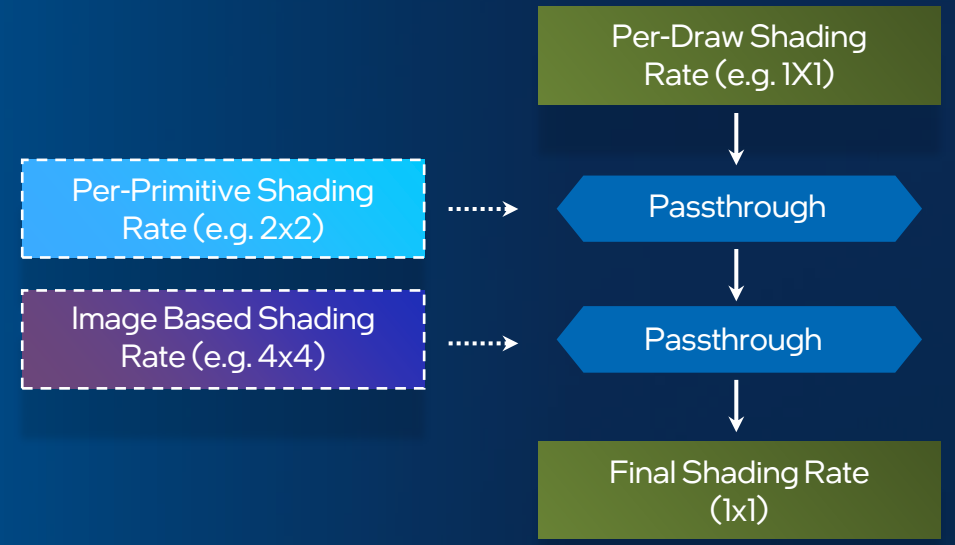
VALAR Image Based Override Combiner




```
D3D12_SHADING_RATE_COMBINER shadingRateCombinerVALAR[2] = {
  D3D12_SHADING_RATE_COMBINER_PASSTHROUGH,
  D3D12_SHADING_RATE_COMBINER_OVERRIDE
};

commandList->RSSetShadingRate(
  D3D12_SHADING_RATE_1X1, shadingRateCombinerVALAR);
```

Manual-Alpha Test Passthrough Combiner




```
D3D12_SHADING_RATE_COMBINER shadingRateCombinerHero[2] = {
  D3D12_SHADING_RATE_COMBINER_PASSTHROUGH,
  D3D12_SHADING_RATE_COMBINER_PASSTHROUGH
};

commandList->RSSetShadingRate(
  D3D12_SHADING_RATE_1X1, shadingRateCombinerHero);
```

VALAR DISABLED
3840x2160 Quality 5

*Images Courtesy of Blizzard Entertainment, Inc.

VALAR STANDARD
3840x2160 Quality 5

Sensitivity
Threshold: **0.31**

Quarter Rate Shading
Modifier: **2.13**

Environment
Luma: **0.05**

Weber-Fechner Mode:
Disabled

*Images Courtesy of Blizzard Entertainment, Inc.

VALAR AGGRESSIVE
3840x2160 Quality 5

Sensitivity
Threshold: **0.50**

Quarter Rate Shading
Modifier: **2.13**

Environment
Luma: **0.05**

Weber-Fechner Mode:
Disabled

*Images Courtesy of Blizzard Entertainment, Inc.

Side-By-Side Image Comparison

VALAR DISABLED

Resolution: 3840x2160
Quality Level: 5
Zoom: 100%

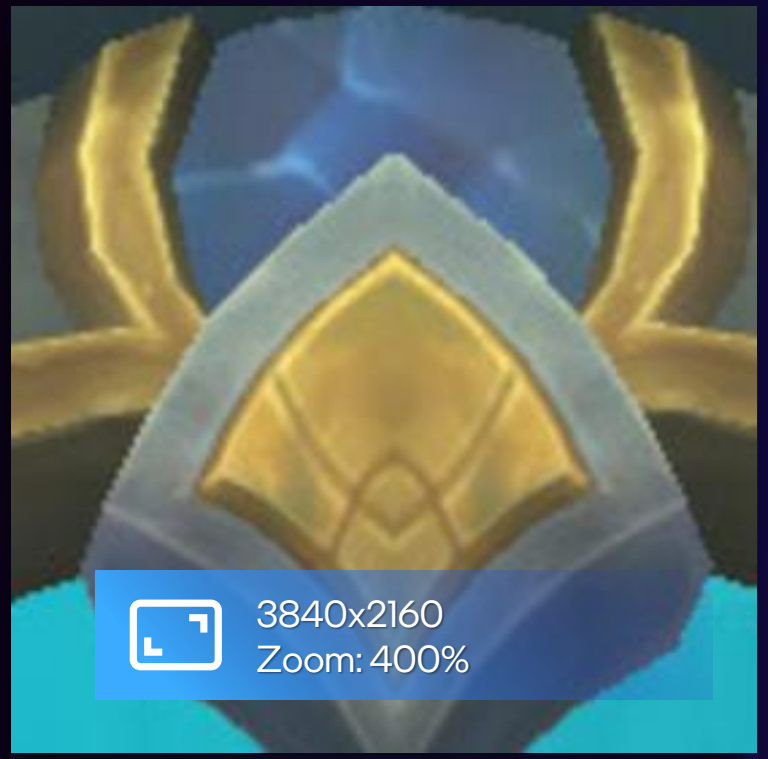
VALAR STANDARD

Sensitivity Threshold: **0.31**
Quarter Rate Shading Modifier: **2.13**
Environment Luma: **0.05**
Weber-Fechner Mode: **Disabled**

VALAR AGGRESSIVE

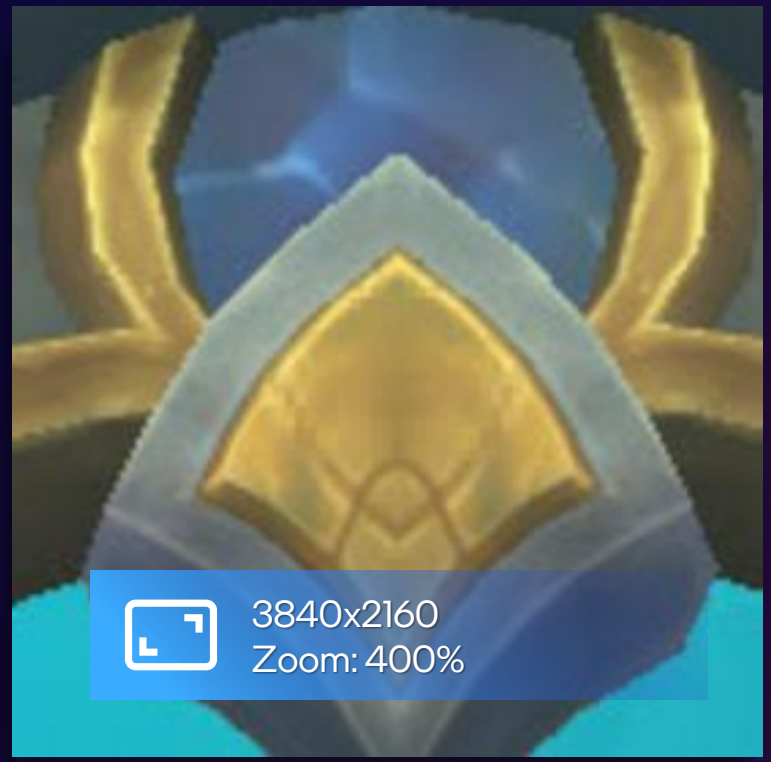
Sensitivity Threshold: **0.50**
Quarter Rate Shading Modifier: **2.13**
Environment Luma: **0.05**
Weber-Fechner Mode: **Disabled**

Side-By-Side Image Comparison



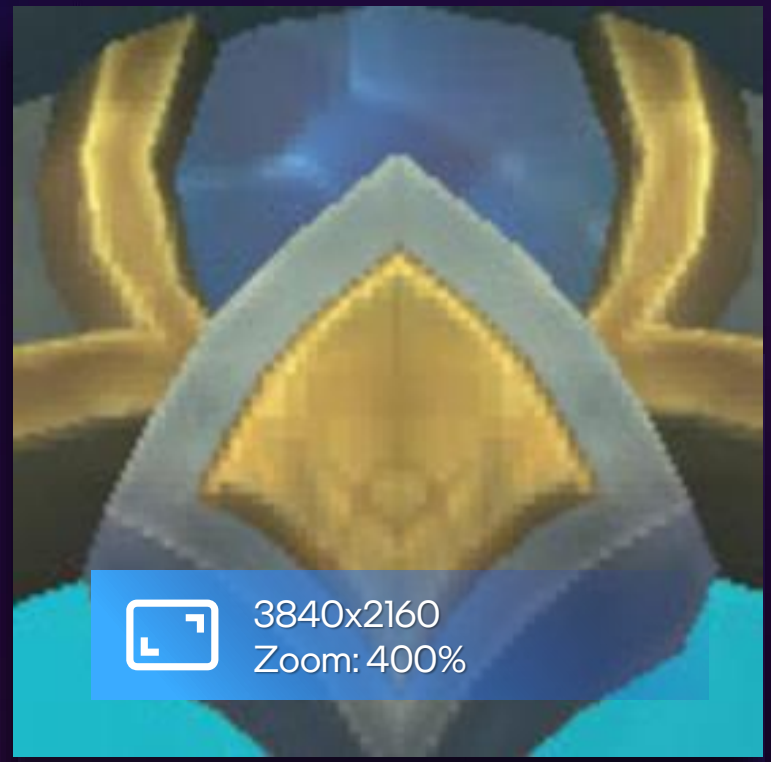
VALAR DISABLED

Resolution: 3840x2160
Quality Level: 5
Zoom: 400%



VALAR STANDARD

Sensitivity Threshold: **0.31**
Quarter Rate Shading Modifier: **2.13**
Environment Luma: **0.05**
Weber-Fechner Mode: **Disabled**



VALAR AGGRESSIVE

Sensitivity Threshold: **0.50**
Quarter Rate Shading Modifier: **2.13**
Environment Luma: **0.05**
Weber-Fechner Mode: **Disabled**

*Images Courtesy of Blizzard Entertainment, Inc.

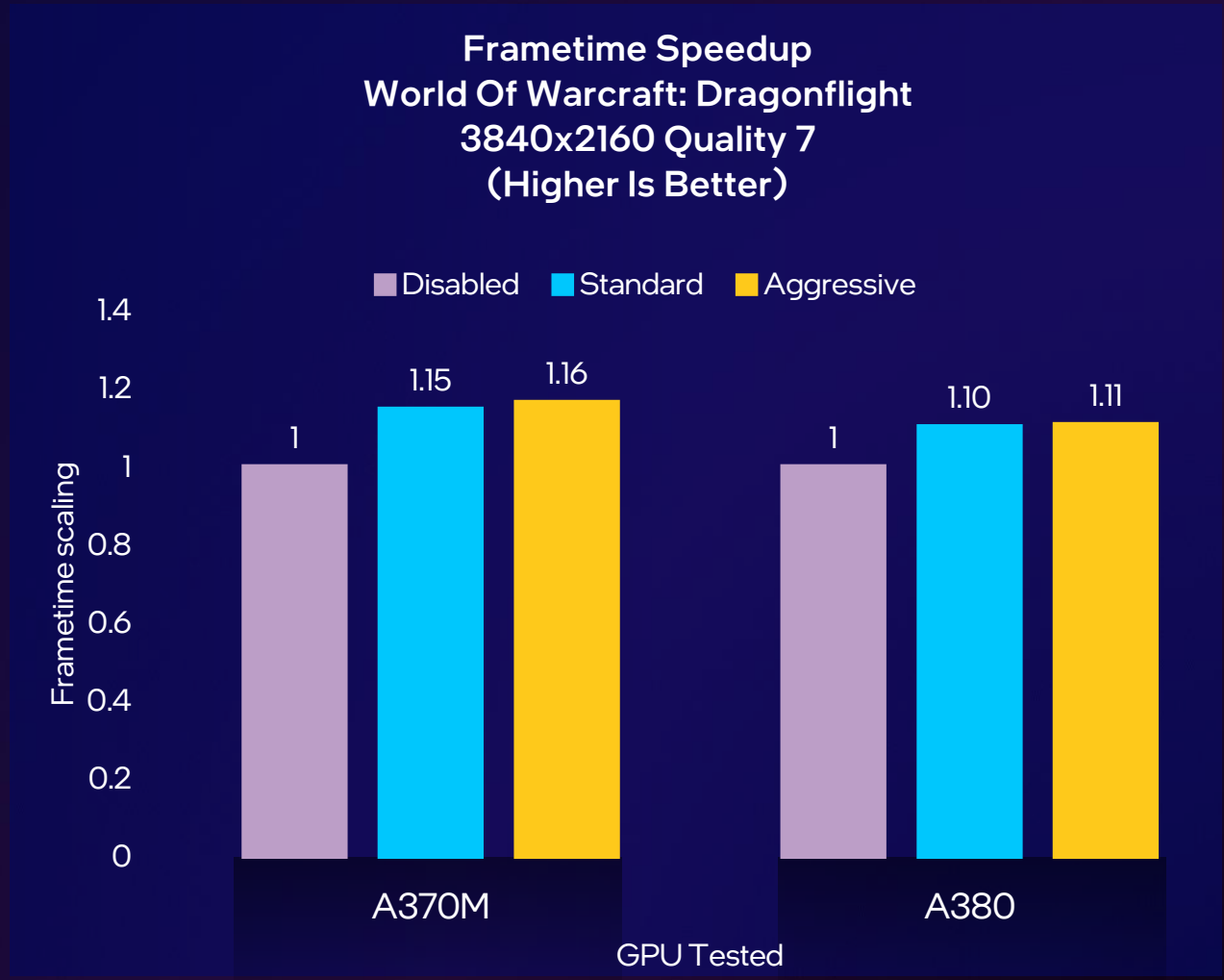
Performance Speedups

Test Configuration

- Standard Sensitivity: 0.31
- Aggressive Sensitivity: 0.50
- Quarter Rate Shading Modifier: 2.13
- Environment Luminance: 0.05
- Resolution: 3840x2160
- Quality Level: 7
- Area: Vol'Dun

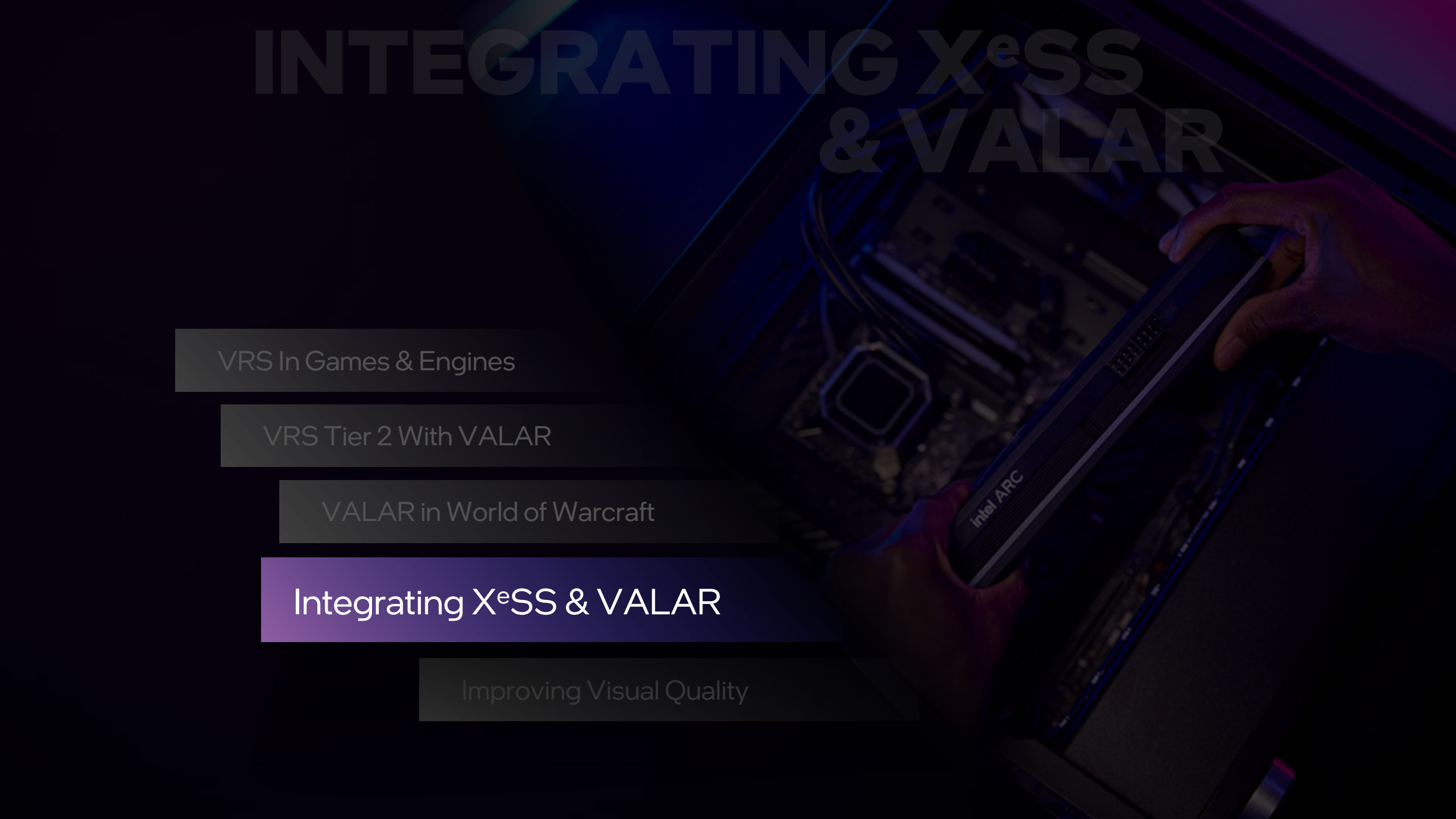
Frametime Scaling

- Intel® Arc A370M: Up to 1.15x to 1.16x
- Intel® Arc A380: Up to 1.10x to 1.11x



*Pre-production software, numbers subject to change.
Arc A370M: 12th Gen Intel(R) Core(TM) i7-12700H - Windows 11 Pro 64-bit (10.0, Build 22621) - 32768MB RAM
Arc A380: 12th Gen Intel(R) Core(TM) i5-12600K - Windows 11 Pro 64-bit (10.0, Build 22621) - 32768MB RAM

INTEGRATING XeSS & VALAR



VRS In Games & Engines

VRS Tier 2 With VALAR

VALAR in World of Warcraft

Integrating XeSS & VALAR

Improving Visual Quality

Integrating XeSS & VALAR

Xe Super-Sampling (XeSS)

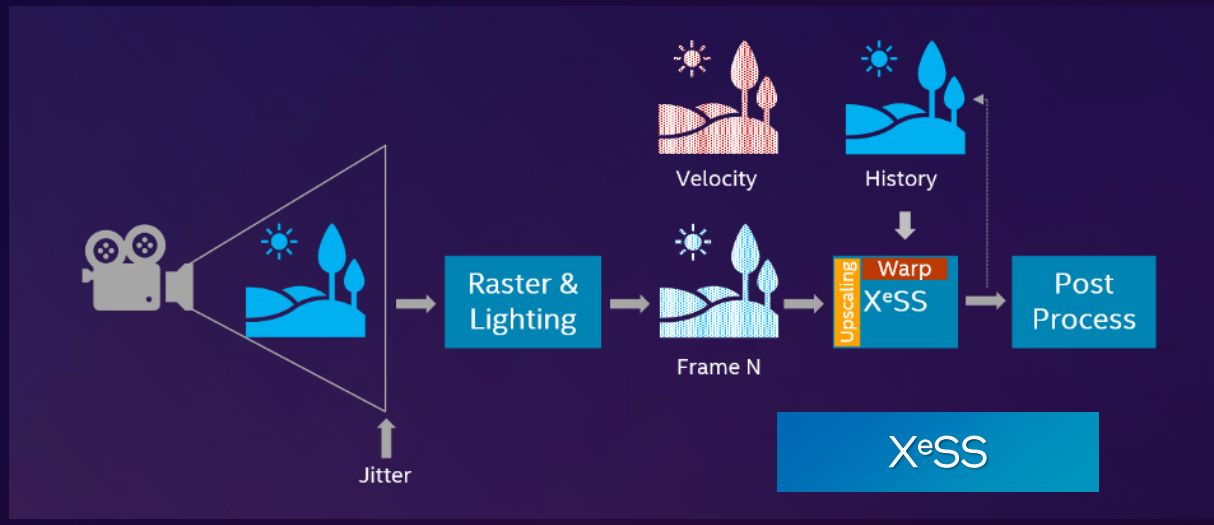
- Machine-Learning Based Super-Sampling
- Temporally Stable Anti-Aliased Output
- Low-Res & High-Res Motion Vectors
- Scaling Factors: Ultra, Quality, Balanced, Performance

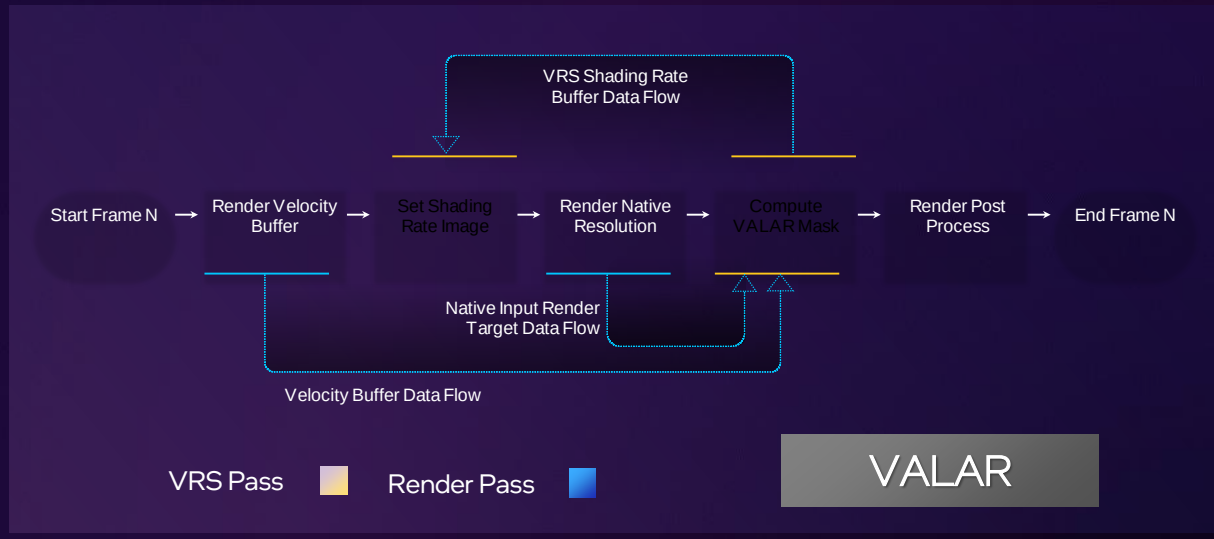
Velocity & Luminance Adaptive Rasterization

- VRS Tier 2 Based Sub-Sampling Technique
- Approximates "Perceptually Lossless" Output
- Low-Res & High-Res Motion Vectors
- Shader Performance Relative To Input Resolution

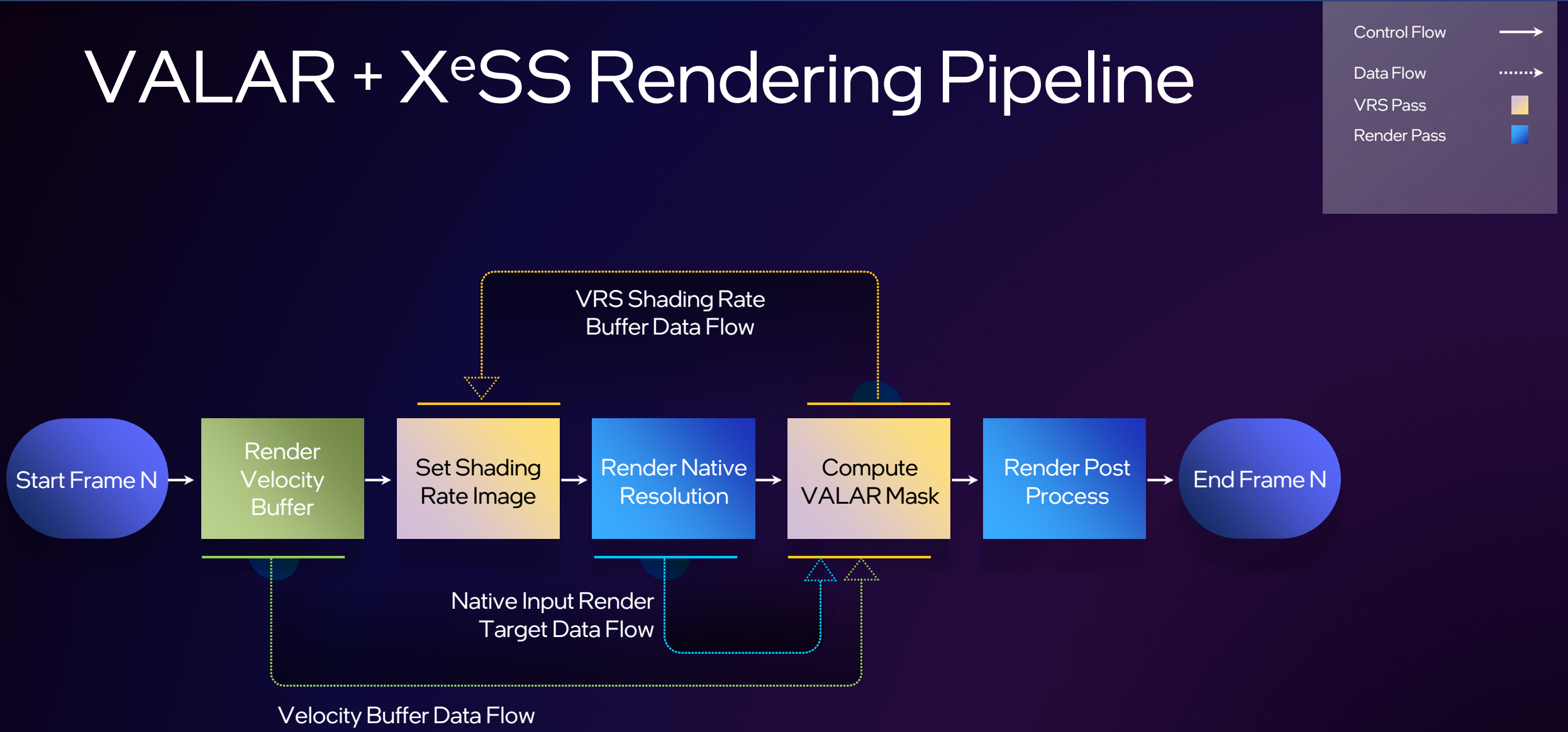
Considerations

- Mapping VRS Tiles to Upscaled Pixels
- Translating Between Coordinate Systems
- Low-Res & High-Res Motion Vectors
- Improving Visual Quality

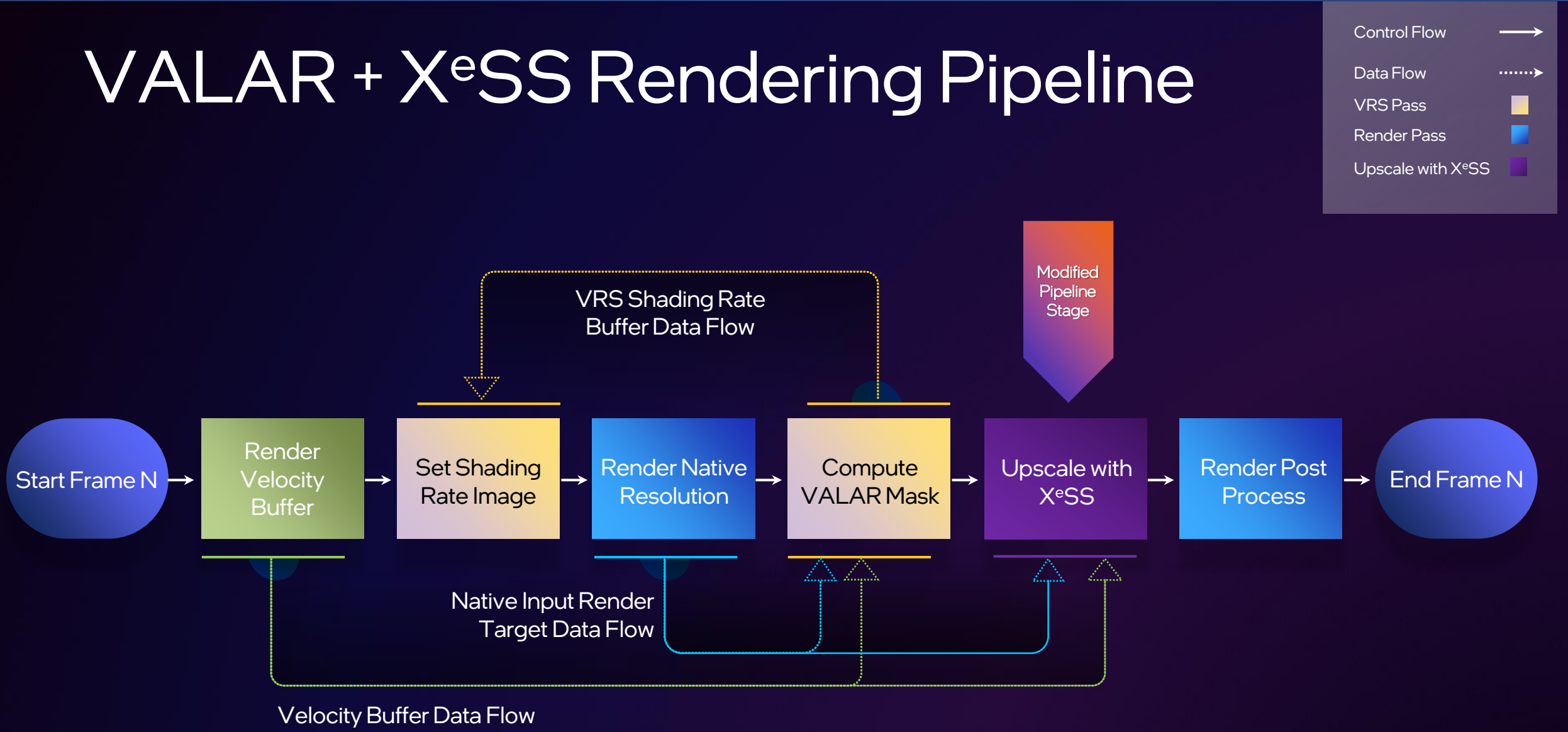




VALAR + XeSS Rendering Pipeline



VALAR + XeSS Rendering Pipeline



VRS Tile Sizes & Scaled Resolutions

Tile Sizes Defined by GPU Hardware

- $T = \begin{cases} 8, & \text{Intel;} \\ 16, & \text{Other.} \end{cases}$

Check Hardware Features

Number of Tiles Relative To Input Resolution

- $N_i = \left(\frac{w_i}{T}\right)\left(\frac{h_i}{T}\right)$
- $N_{2k} = \left(\frac{2560}{8}\right)\left(\frac{1440}{8}\right) = (320 \cdot 180) = \mathbf{57,600}$
- $N_{4k} = \left(\frac{3840}{8}\right)\left(\frac{1600}{8}\right) = (480 \cdot 200) = \mathbf{96,000}$

Common Scaling Factors for XeSS

- $S = \begin{cases} \mathbf{1.3}, & Ultra; \\ \mathbf{1.5}, & Quality; \\ \mathbf{1.7}, & Balanced; \\ \mathbf{2.0}, & Performance. \end{cases}$

Input And Upscaled Tile Sizes Are Not Equal

- $T_i = T^2$
- $T_u = (S \cdot T)^2$
- $T_u \geq T_i$

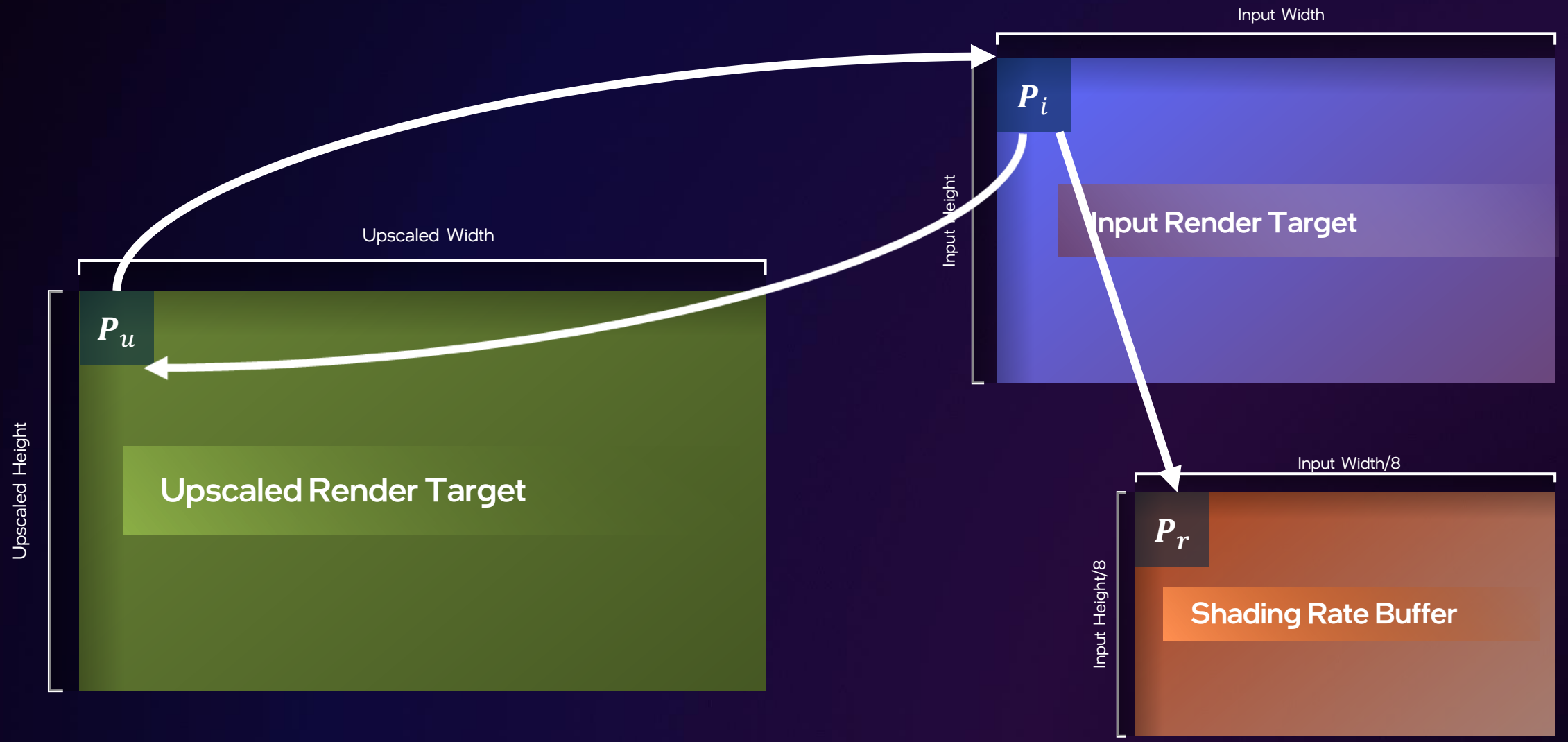
Number of Input Resolution Pixels Per Tile

- $P_{intel} = 8^2 = \mathbf{64}$
- $P_{other} = 16^2 = \mathbf{256}$

Number of Upscaled Pixels Per Tile

- $P_{ultra} = (1.3 \cdot 8)^2 \approx \mathbf{108}$
- $P_{quality} = (1.5 \cdot 8)^2 = \mathbf{144}$
- $P_{balanced} = (1.7 \cdot 8)^2 \approx \mathbf{185}$
- $P_{performance} = (2 \cdot 8)^2 = \mathbf{256}$

Pixel Coordinate Conversions



Scaling Pixel Coordinates

Compute The Scaling Ratio

$$S = \frac{I}{U}$$

2D Vectors

Convert From Upscaled To Input Resolution

$$P_i = S \cdot P_u$$

Convert From Input Resolution to VRS Tile

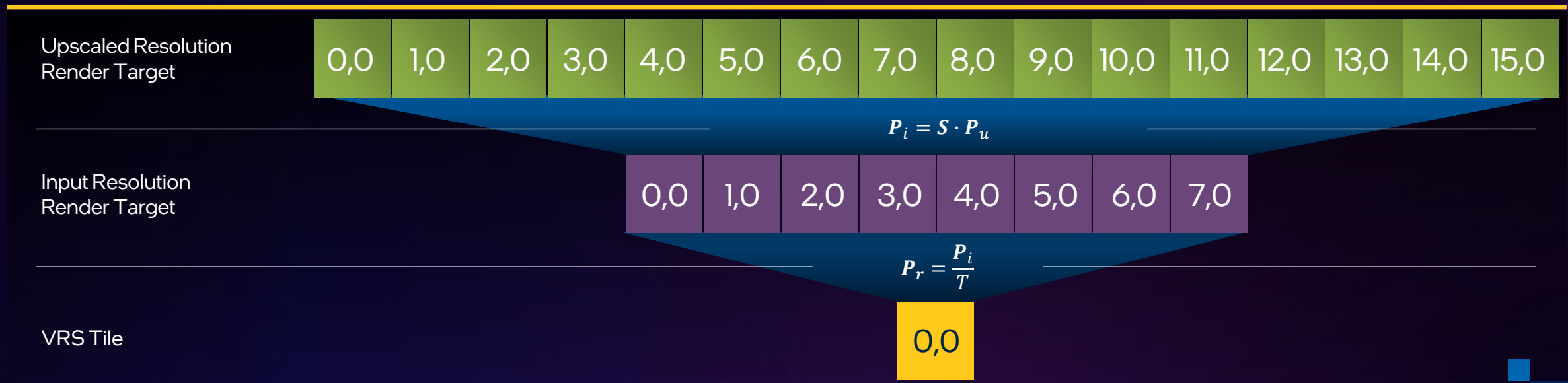
$$P_r = \frac{P_i}{T}$$

HW Tile Size

Convert From Input To Upscaled Resolution

$$P_u = \frac{P_i}{S}$$

$P_u \in P_u \ldots P_u + (T - 1)$



Low-Res & High-Res Motion Vectors

Low-Res Motion Vectors

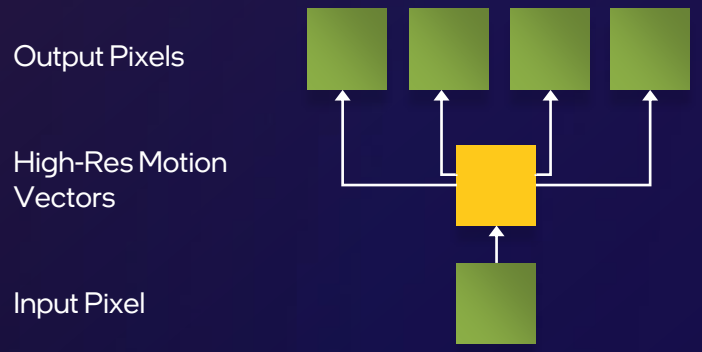
- Input And Motion Vector Resolution Are Equal
- Works out-of-the-box With VALAR
- Jittered Motion Vectors Can Improve Stability

High-Res Motion Vectors

- Output And Motion Vector Resolution Are Equal
- High-Res Motion Vector Support Added To VALAR
- Jittered Motion Vectors Can Improve Stability

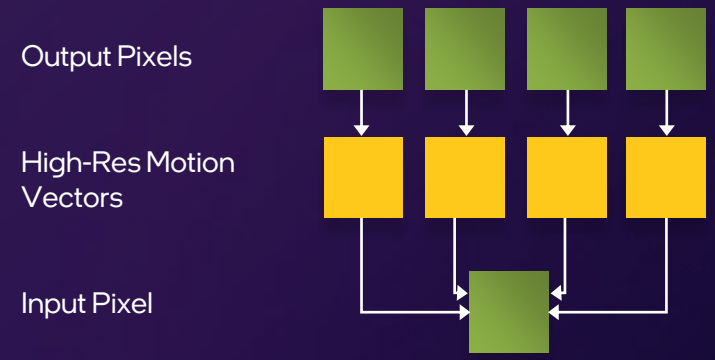
Low-Res Coordinate Conversion

- One Motion Vector to Many Output Pixels
- One-To-One With Input Pixels
- $P_i = S \cdot P_u$



High-Res Coordinate Conversion

- Many Motion Vectors to One Input Pixel
- Sampling Pattern or Min/Max/Median/Average
- $P_u = \frac{P_i}{S}$ where $P_u \in P_u \ldots P_u + (T - 1)$



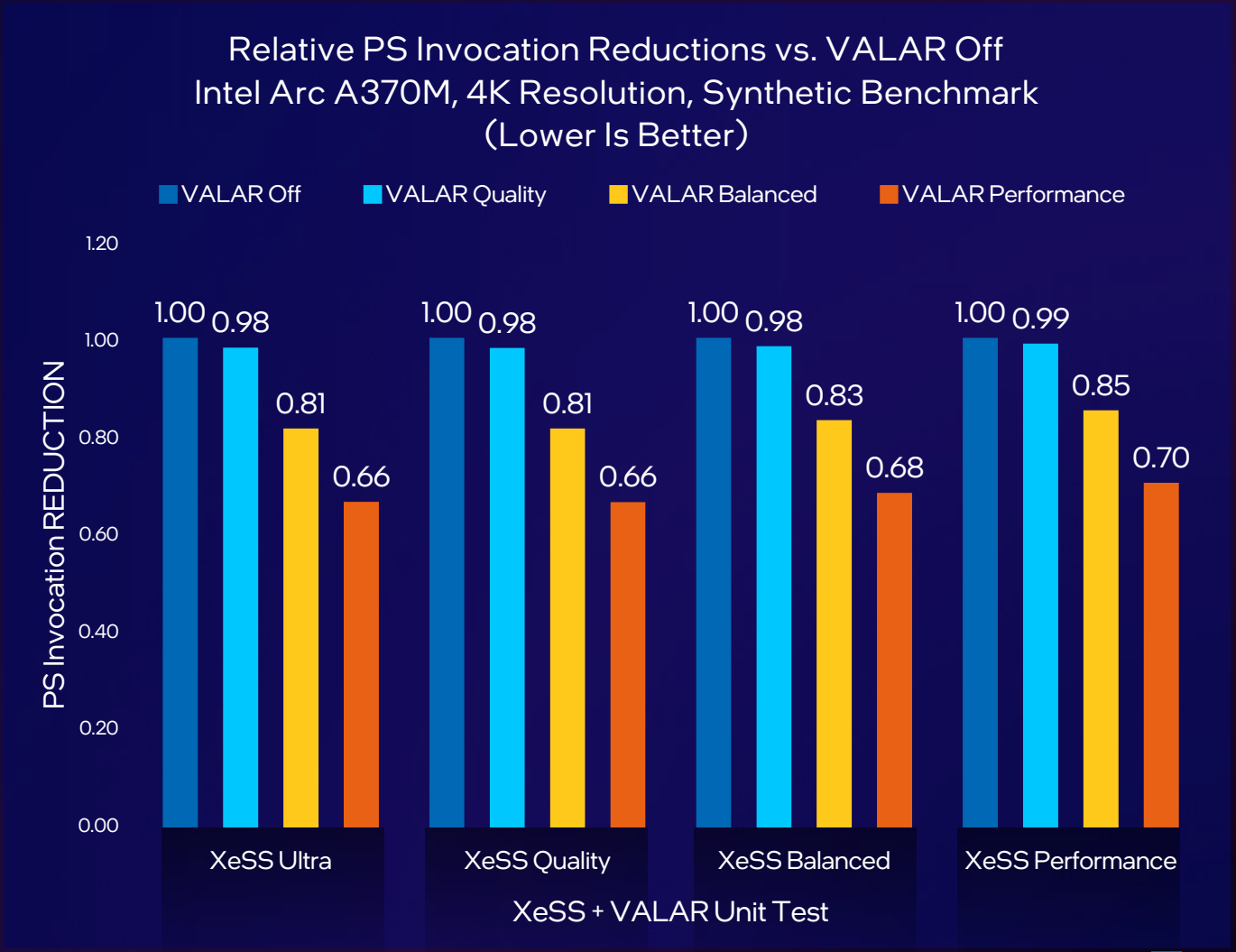
XeSS + VALAR Pixel Shader Invocations

Experimental Setup

- XeSS Mode + VALAR Mode
- Relative Reduction in PS Invocations
- Versus Control: XeSS Mode + VALAR OFF
- VALAR Quality: Sensitivity = 0.25
- VALAR Balanced: Sensitivity = 0.50
- VALAR Performance: Sensitivity = 0.75

Observations

- PS Invocations Relative to VALAR Sensitivity
- VALAR Quality: ~1-2% Reduction
- VALAR Balanced: ~15-20% Reduction
- VALAR Performance: ~30-34% Reduction



***Pre-production software, numbers subject to change.**
12th Gen Intel(R) Core(TM) i7-12700H - Windows 11 Pro 64-bit (10.0, Build 22621) - 32768MB RAM

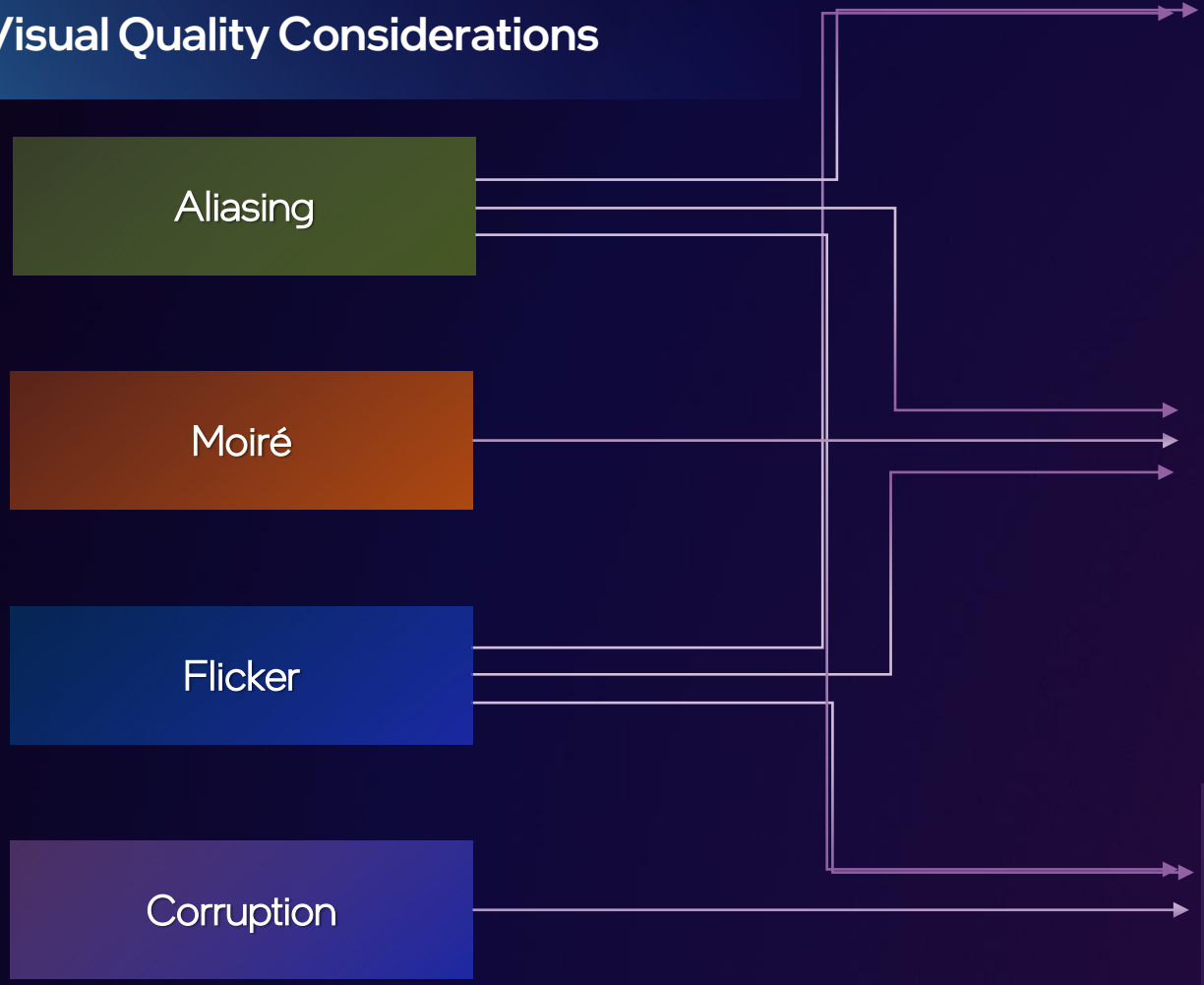
IMPROVING VISUAL QUALITY

- VRS In Games & Engines
- VRS Tier 2 With VALAR
- VALAR in World of Warcraft
- Integrating XeSS & VALAR
- **Improving Visual Quality**

Visual Quality Improvement Strategies

Visual Quality Considerations

- Aliasing
- Moiré
- Flicker
- Corruption



VALAR Quality Tuning

- Parameter Tuning
- 2x2 Only Mode
- Weber-Fechner Mode
- Dynamic Threshold
- Relaxed Mode

XeSS Quality Tuning

- Quality Modes
- Mip-Biasing
- Sharpening
- Jittered Motion Vectors

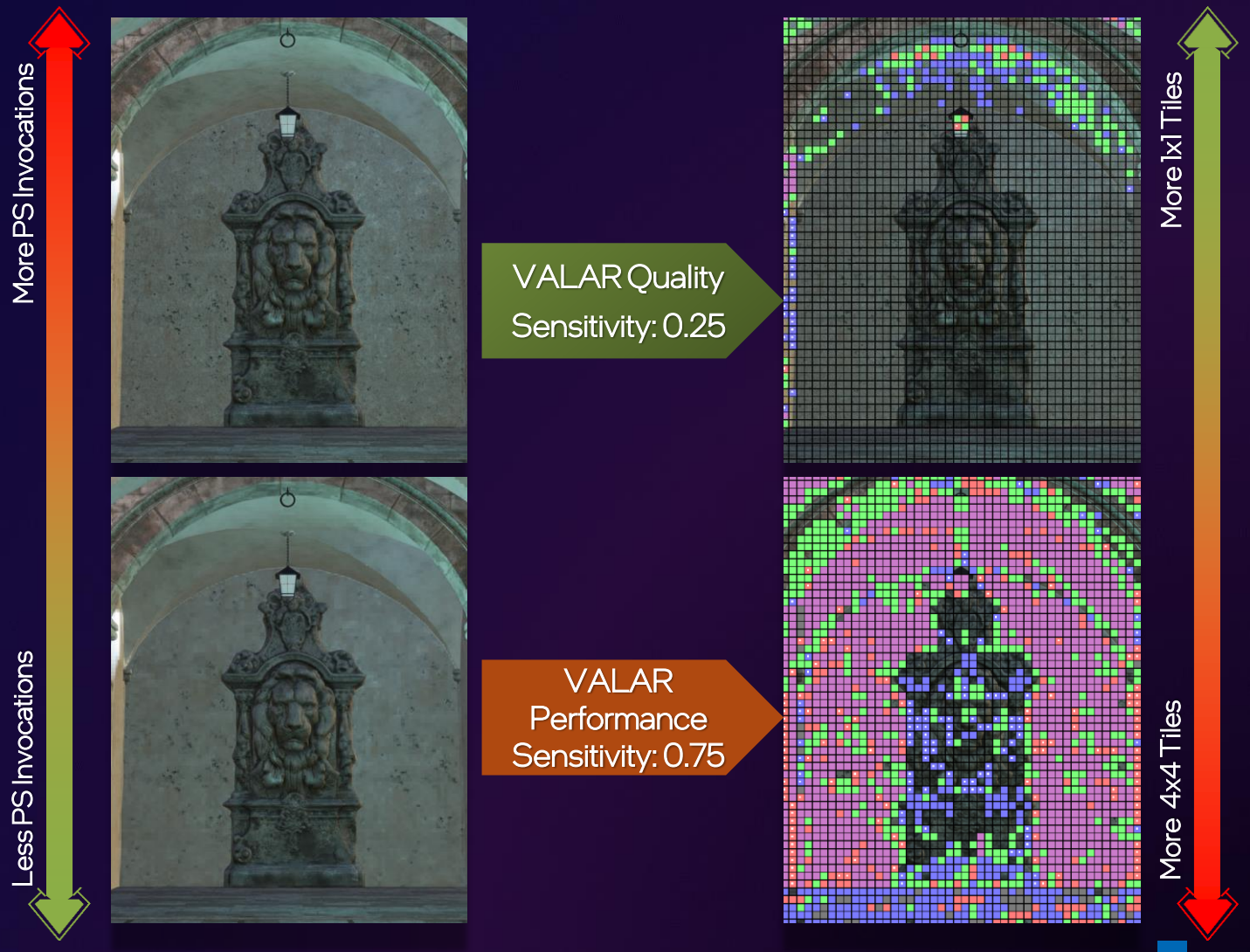
API Quality Tuning

- VRS Combiners
- Multiple Command Lists

VALAR Quality Tuning

Sensitivity Threshold (T)

- Primary Threshold For JND
- Consider Quality Modes
  - Quality: 0.25
  - Balanced: 0.50
  - Performance: 0.75



$$JND = \boldsymbol{T} \cdot (\bar{L}_t + E)$$

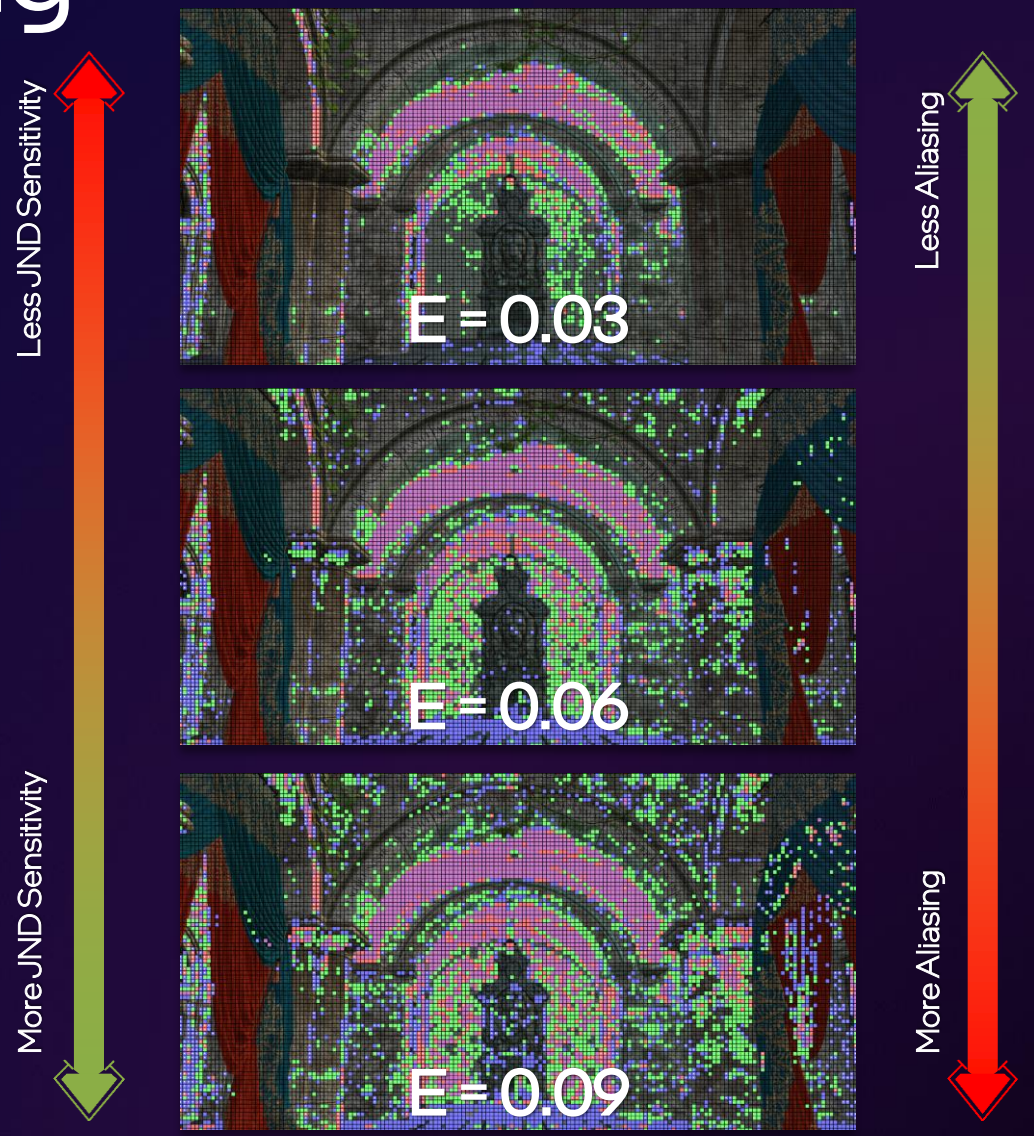
VALAR Quality Tuning

Sensitivity Threshold (T)

- Primary Threshold For JND
- Consider Quality Modes
  - Quality: 0.25
  - Balanced: 0.50
  - Performance: 0.75

Environment Luminance (E)

- Added To Average Luma in JND
- Use In Low Light Conditions
- Use GI Value [optional]



$$JND = \boldsymbol{T} \cdot (\bar{L}_t + E)$$

VALAR Quality Tuning

Sensitivity Threshold (T)

- Primary Threshold For JND
- Consider Quality Modes
  - Quality: 0.25
  - Balanced: 0.50
  - Performance: 0.75

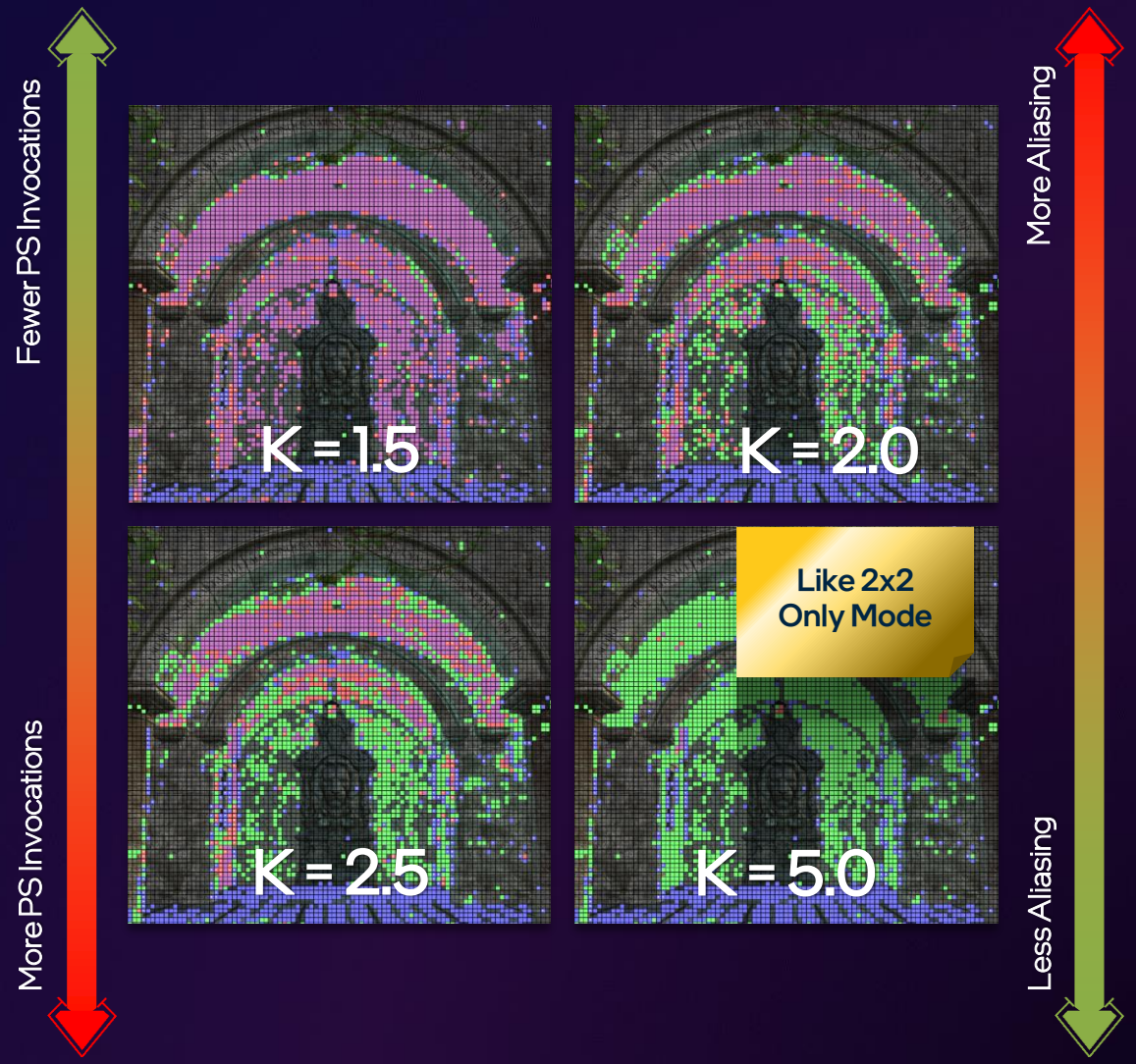
Environment Luminance (E)

- Added To Average Luma in JND
- Use In Low Light Conditions
- Use GI Value [optional]

Quarter Rate Shading Modifier (K)

- Controls 2x4, 4x2, and 4x4
- Part of Luma/Velocity MSE
- Recommended: 2.13 [Yang, 2019]
- High Values Decrease Quarter Rate

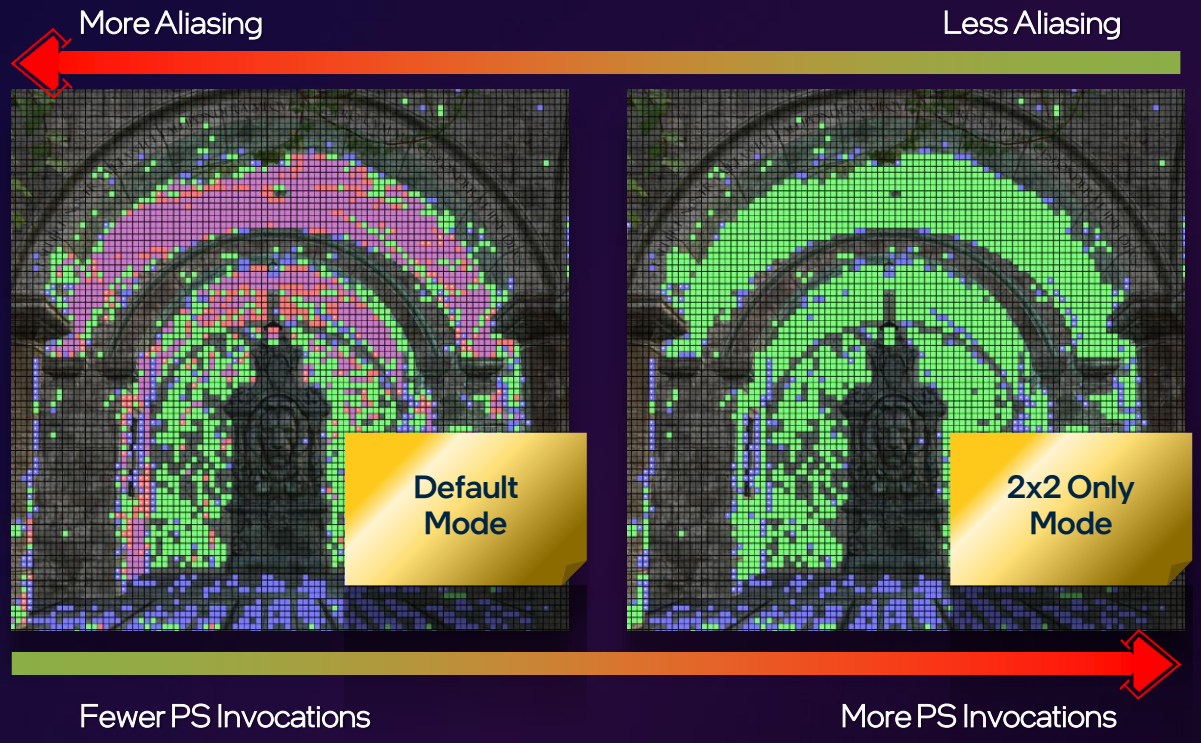
$$JND = \boldsymbol{T} \cdot (\bar{L}_t + E)$$



VALAR Optional Modes

2x2 Only Mode

- Disables 2x4, 4x2, and 4x4
- Increases 2x2 Tiles
- Reduces Aliasing In Coarse Pixels



VALAR Optional Modes

2x2 Only Mode

- Disables 2x4, 4x2, and 4x4
- Increases 2x2 Tiles
- Reduces Aliasing In Coarse Pixels

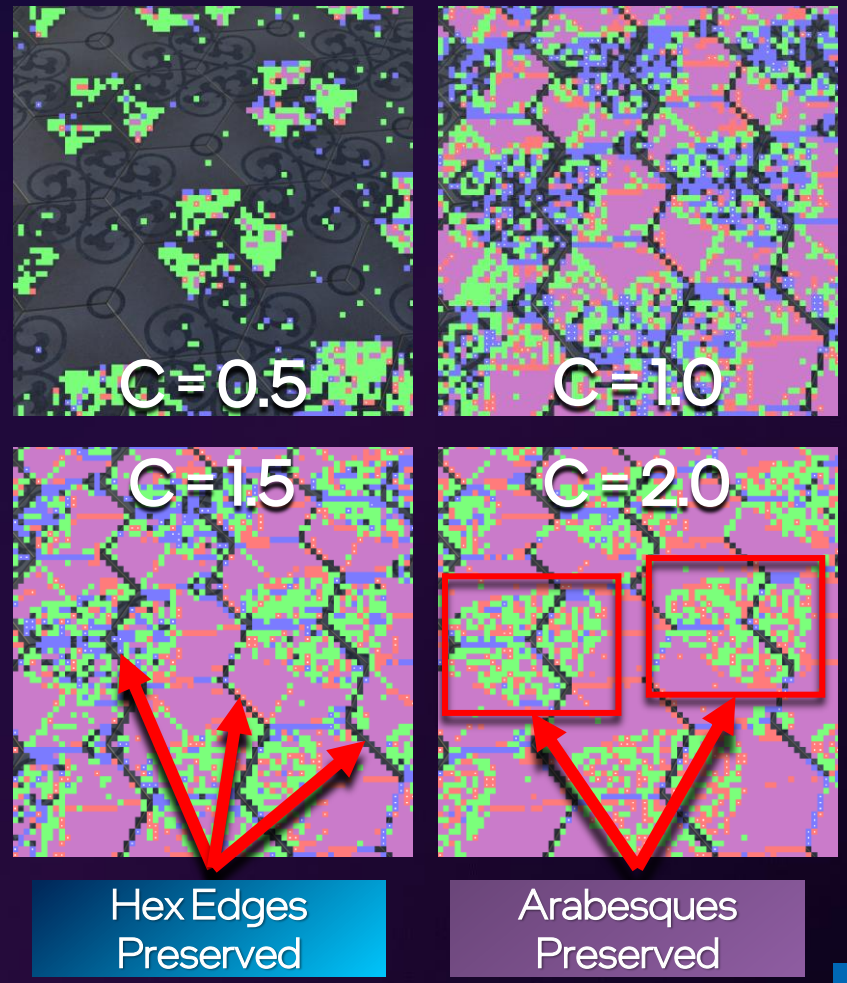
Weber-Fechner Mode

- Improves Precision of MSE
- Reads Neighborhood Luminance
- Weber-Fechner Constant (C)
- Tunable Precision with C

Bistro Scene

More Conservative

Default Setting



Hex Edges Preserved

Arabesques Preserved

VALAR Optional Modes

2x2 Only Mode

- Disables 2x4, 4x2, and 4x4
- Increases 2x2 Tiles
- Reduces Aliasing In Coarse Pixels

Weber-Fechner Mode

- Improves Precision of MSE
- Reads Neighborhood Luminance
- Weber-Fechner Constant (C)
- Tunable Precision with C

Dynamic Threshold

- Enable / Disable VRS Techniques
- Auto-Adjust VALAR Sensitivity
- When GPU Bound
- Exceeding Target Framerate

GPU Bound Toggle

$$VALAR = \begin{cases} ON, & \bar{t}_{gpu} \geq \bar{t}_{cpu} - (\sim 0.05 \cdot \bar{t}_{cpu}); \\ OFF, & otherwise. \end{cases}$$

Exceeding Target Framerate

$$T = \frac{\bar{t}_{gpu}}{1000 \cdot t_{fps}} \; where \; \boldsymbol{T} \; is \; the \; sensitivity \; threshold$$

Relaxed Mode & Relaxed Mode Plus

Relaxed Mode

- $VALAR = \begin{cases} DISABLED, & C_N - C_{N-1} = 0; \\ ENABLED, & otherwise. \end{cases}$

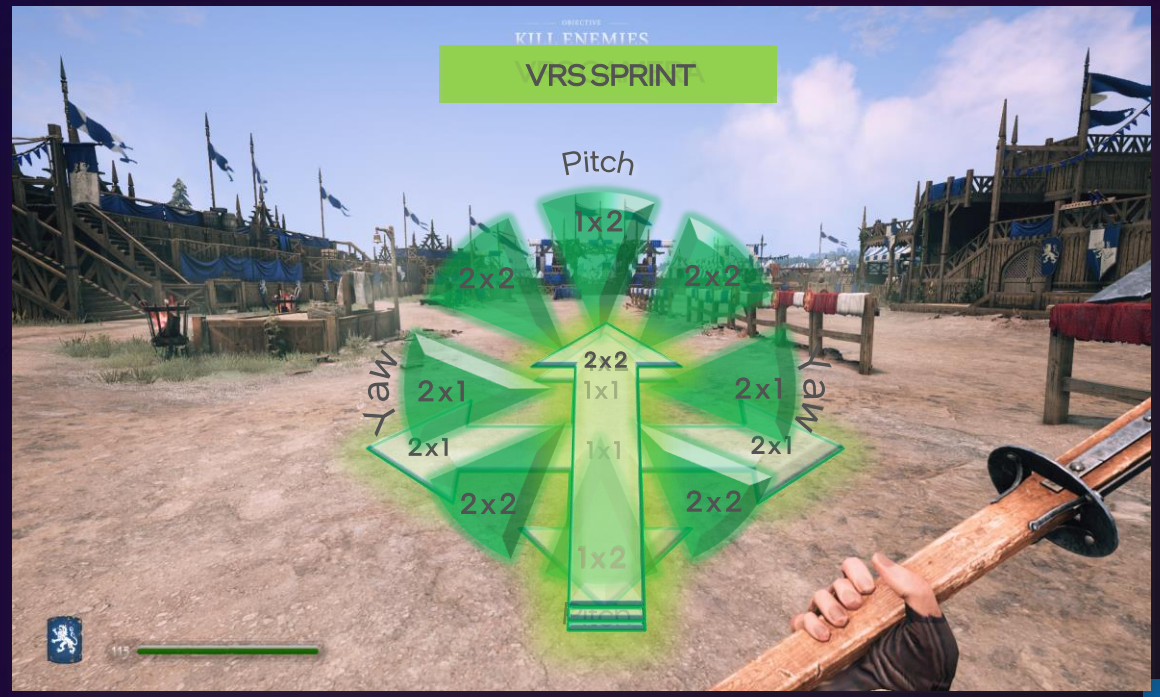
Relaxed Mode Plus

- $R_x = \begin{cases} R_2, & \leftarrow C \rightarrow; \\ R_1, & otherwise. \end{cases}$
- $R_y = \begin{cases} R_2, & \uparrow C \downarrow; \\ R_1, & otherwise. \end{cases}$
- $R_{xy} = \begin{cases} R_2, & \swarrow C \nearrow \; or \; \nwarrow C \searrow; \\ R_1, & otherwise. \end{cases}$

Chivalry II by Tripwire*

- Released in 2021
- VRS Tier 1 Relaxed Mode Plus
- Enabled VRS When Camera or Player Moves

Motion & Motion Blur Obscure Aliasing



*Images Courtesy of Tripwire Interactive

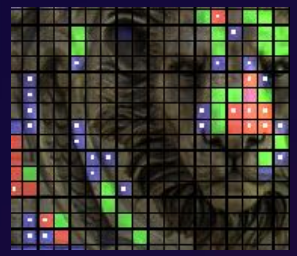
XeSS Quality Tuning

Quality Modes

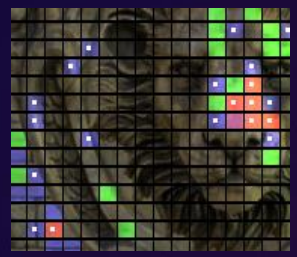
- Lower Scaling Factors May Reduce Aliasing
- Lower Scaling Factors May Reduce Flicker
- Lower Scaling Factors Decrease VRS Coverage
- Higher Scaling Factors Improve Performance

More VRS Coverage

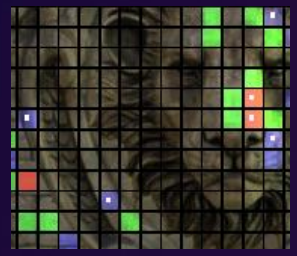
Less VRS Coverage



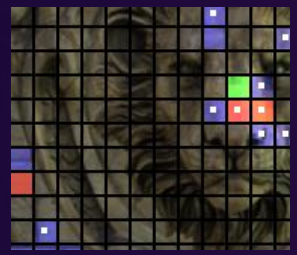
1080p - XeSS Ultra - VALAR Balanced
~18x15 Tiles → ~64 Shaded Tiles



1080p - XeSS Quality - VALAR Balanced
~16x14 Tiles → ~40 Shaded Tiles



1080p - XeSS Balanced - VALAR Balanced
~13x12 Tiles → ~20 Shaded Tiles



1080p - XeSS Perf - VALAR Balanced
~12x10 Tiles → ~13 Shaded Tiles

Higher VRS Precision

Lower VRS Precision

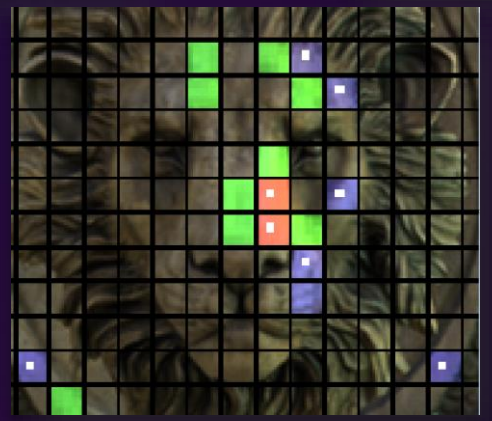
XeSS Quality Tuning

Quality Modes

- Lower Scaling Factors May Reduce Aliasing
- Lower Scaling Factors May Reduce Flicker
- Lower Scaling Factors Decrease VRS Coverage
- Higher Scaling Factors Improve Performance

Sharpening

- Blurs Or Sharpens Upscaled Output
- Helps Hide Aliasing From VALAR



XeSS Quality Tuning

Quality Modes

- Lower Scaling Factors May Reduce Aliasing
- Lower Scaling Factors May Reduce Flicker
- Lower Scaling Factors Decrease VRS Coverage
- Higher Scaling Factors Improve Performance

Sharpening

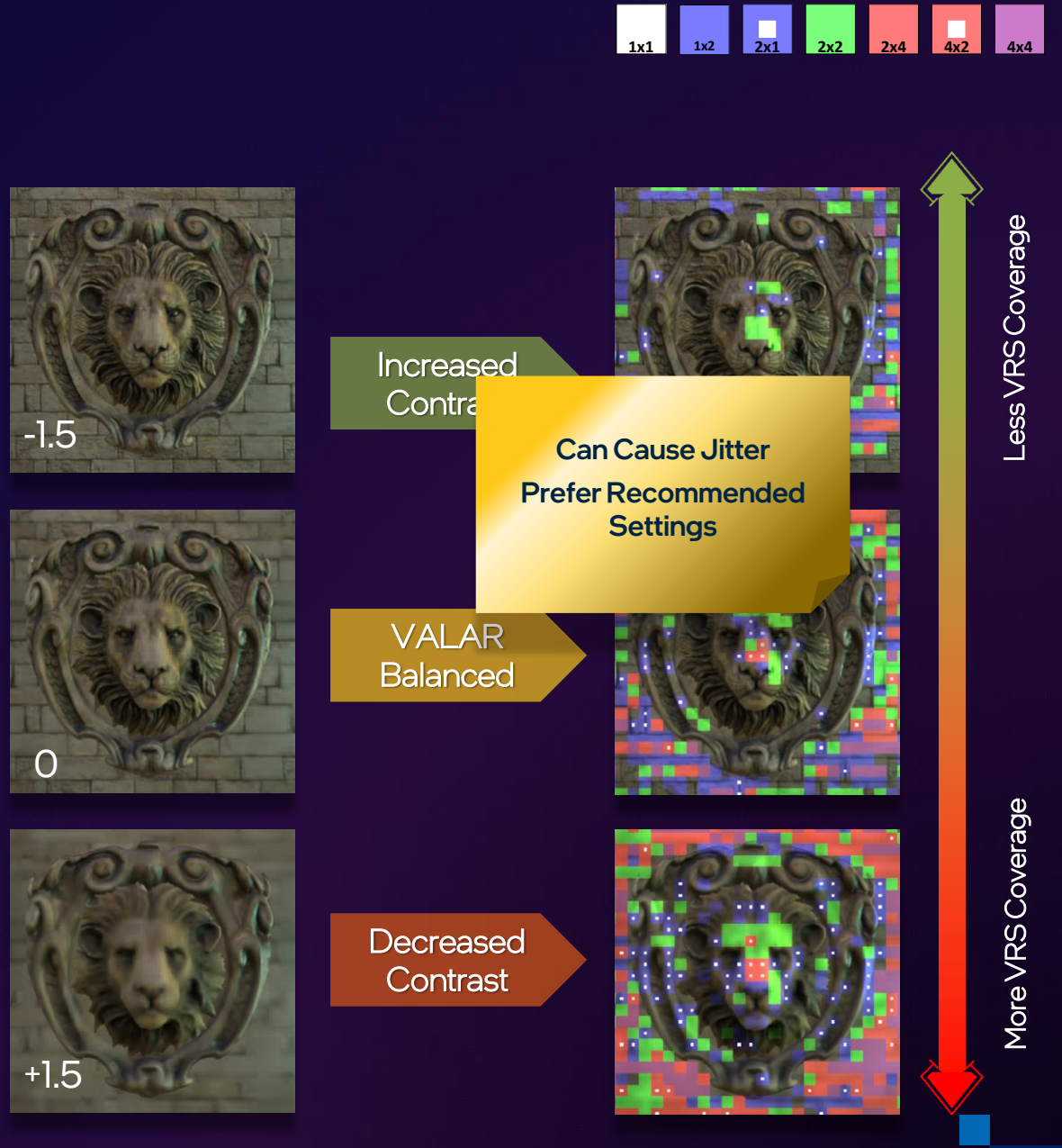
- Blurs Or Sharpens Upscaled Output
- Helps Hide Aliasing From VALAR

Mip-Biasing

- Selectively Biases Mip-Levels
- Helps Control Moiré From XeSS
- Contrast Affects VALAR Sensitivity

Jittered Motion Vectors

- Can Help Reduce Flickering

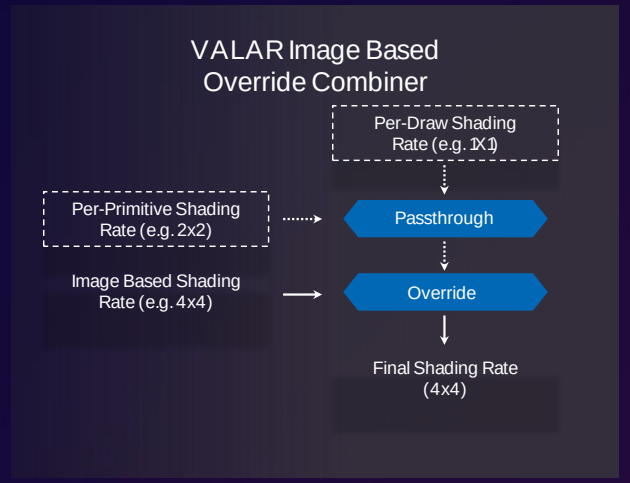


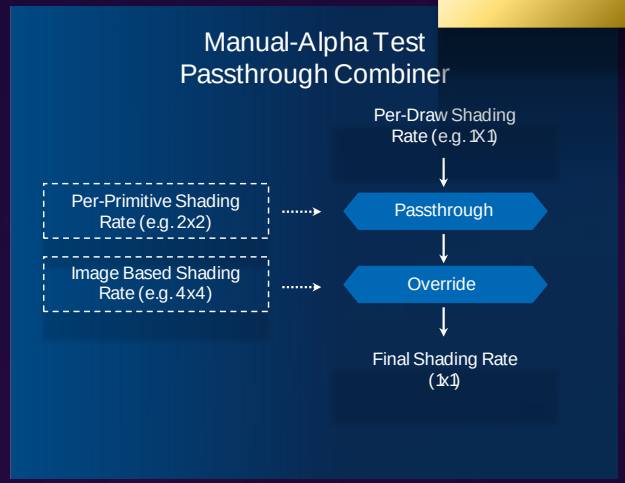
API Quality Tuning

Used in World of Warcraft

VRS Combiners

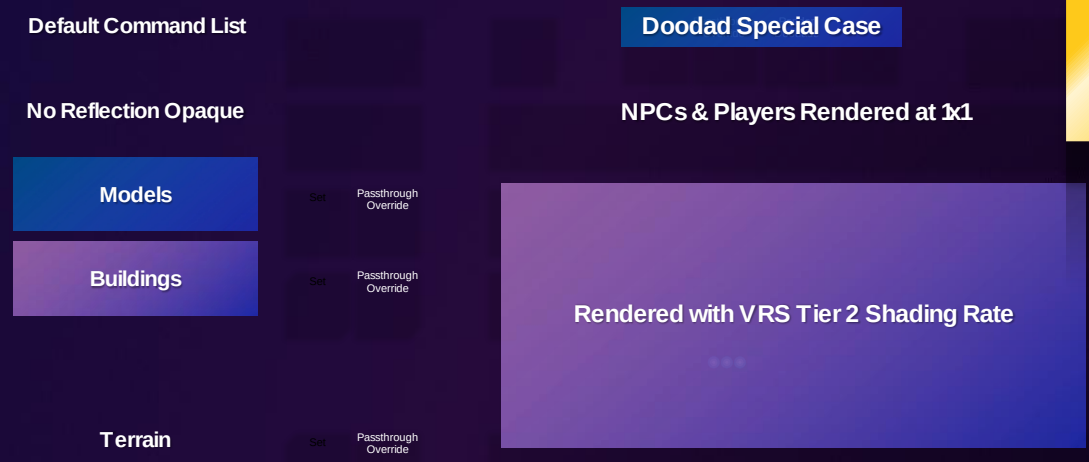
- Passthrough-Passthrough
  - ‘Hero’ Assets
  - Flickering (VRS)
  - Visual Corruption
- Passthrough-Override
  - Image Mask Overrides Tier 1





Multiple Command Lists

- High Quality Command List
  - ‘Hero’ Assets
  - Flickering (VRS)
  - Visual Corruption
- VRS Tier 2 Command List
  - Multiple VRS Image Masks



Used in World of Warcraft

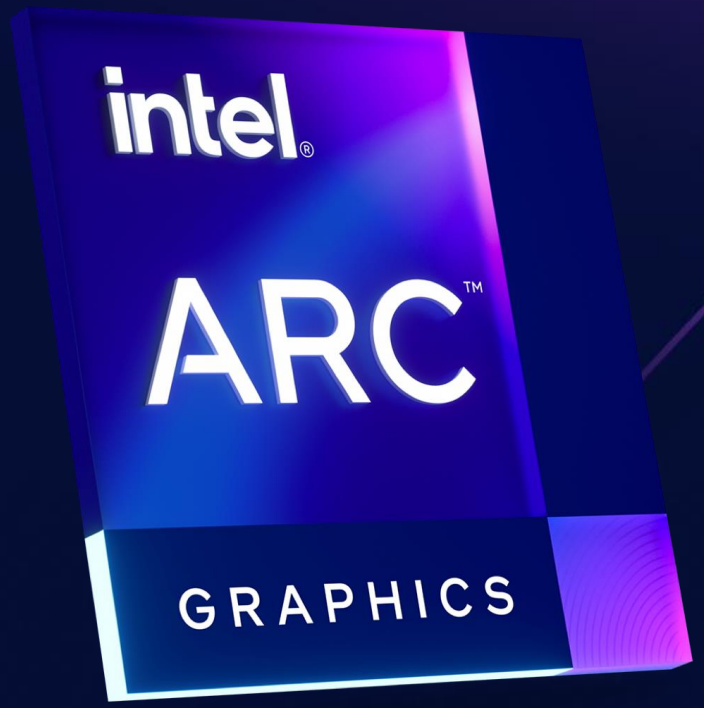
Acknowledgements

Adam Kunka
Adam Lake
Laura Reznikov
Meghan Weicht
Aria Kraft
Anil Alston
Kai Wang
Dmitry Kozlov
Andreas Weinmann
Gilles Brossard
Eric Hoekstra
Karthik Vaidyanathan
Stephen Junkins
Mike Burrows
Leigh Davies
Ethan Davis
Phillip Gerasimov
Alex Kharlamov
Mikio Sakemoto
And More...

BLIZZARD ENTERTAINMENT
Gabriel Belanger
Paul DeBerry
Max Rahm
John Lytle
Blizzard Compat Lab

Microsoft
Claire Andrews
Shawn Hargreaves

intel ARC



Thank you

Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex (graphics and accelerators).

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. No product or component can be absolutely secure.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel technologies may require enabled hardware, software or service activation.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Statements that refer to future plans or expectations are forward-looking statements. These statements are based on current expectations and involve many risks and uncertainties that could cause actual results to differ materially from those expressed or implied in such statements. For more information on the factors that could cause actual results to differ materially, see our most recent earnings release and SEC filings at www.intc.com.


intel


intel
ARC