Visually Lossless Content and Motion Adaptive Shading in

Games

LEI YANG, DMITRY ZHDAN, EMMETT KILGARIFF, ERIC B. LUM, YUBO ZHANG,

MATTHEW JOHNSON, and HENRIK RYDGÅRD, NVIDIA Corporation

The goal of this work was to reduce usage of GPU, but present similar or indistinguishable quality of generated image.They used half, and quarter rate shading. Usage of motion to reduce probability of this reduction in clarity been seen by user. This work already has been used in two commercial projects.

# 1 INTRODACTION

In last year resolution and expected frame rate has grown more that power of GPU. As well as existing of mobile gaming platform has increase demand of more power efficient solution to this problem.

Now exist huge variety of technics to lower computation complexity of shading.

Authors present a Variable Rate Shading (VRS), who’s main goal is to deliver almost the same result with less work.

The algorithms contains of three stages:

(1) Analysing previous frame;

(2) Calculate possible error per screen tile;

(3) Decide which shading rate to use.

It all done with as low overhead as possible.

# 2 BACKGROUND AND RELATED WORK

**2.1 Variable-Rate Shading**

Before there was many different attempt to lower commutation cost, but all of them require significant change in rendering pipeline.

Coarse Pixel Shading or Variable Rate Shading is a feature of nowadays GPUs. It didn’t change resolution but allow to reduce shading for screen tile of 16 X 16 pixels.

They have implemented algorithm, who can select best shading rate from array of 1X1, 1X2, 2X1,2X2,2X4,4X2 and 4X4 sales per shade in 16X16 tiles.

**2.2 Perception Guided Adaptive Shading**

This idea is not a new one, but previous implementation was to complex to use in realtime rendering or need human to correct error tolerance.So authors hase to come up with better solution to the problem of estimate error between coarse- and fine-shaded image.

Idea of VRS is new because it’ only now we have GPU capable of adjusting shading rate. VRS can found great use in VR application, due to non-linear lens wraps.

Other previous algorithm, which have worked with idea of VRS has taking in concern human eye but now a hardware capabilities so authors has to address that.

**2.3 Perceptual Image Quality in Motion**

All type of LCD panels have common flaw: disability to represent linear motion so it result in our eyes as motion blur. It even more visible on common 60 Hz panel but still present on 120 and 165Hz panel as well.

That type of blur are well studied and is similar to applying 1D filter to the image. It is know as motion blur (cinema like blur). This blur effect can be used to better exploit optimisation opportunities.

# 3 CONTENT ADAPTIVE SHADING

Used loss estimator is simple and effective to compute in spatial domain. Loss is compared to Just-Noticeable-Difference(JND) threshold to decide is it possible to use reduce shading rate.

**3.1 Image Error with Half-Rate Shading**

In this part are shown math behind error estimator. In research the came up with two norms for calculation error amount so they combine them into one.

**3.2 Frequency Domain Analysis**

There are explanation of math behind connection of error estimator and frequency response of box filter implemented by half-rate shading.

**3.3 Quarter-Rate Shading**

Is an extension of previous Half-Rate Shading but with use of different box filter. You also can find explanation of math behind simplification of this calculation.

**3.4 Shading Rate Adaptation with a Perceptually-Correct Threshold**

There are description of how was found threshold to norm of the pixel (luma) value error.

# 4 MOTION ADAPTIVE SHADING

In this section is described how to compute motion adjusted error estimation. As ask option is shown how to approximate motion adaptive shading without evaluation the error estimator.

**4.1 Diminished Error Under Motion Blur**

There are showing with graph that speed of the motion has huge influence on attenuation if high-frequency content. Authors describe how similar technics are applied to calculations of error estimator but now it depends on velocity of the motion rather than average luma value. Because in one frame we can have several moving object we should use velocity of slowest one.

**4.2 Standalone Motion Adaptive Shading**

Authors have shown how motion influence can augment VRS. In some application it can be use as standalone feature so in has use simplified formula to calculate of rate decision criteria.

It can be simplified even more because we mostly know how complex game scene is so we can calculate the threshold in advance so it will depends only on velocity of an object.

# 5 IMPLIMINTATION

Algorithm can be implemented in one or few compute passes that run before rasterisation of a frame.Main goal of the implementation was “do no harm”. So it has to be efficient and on 4K UHD resolution.Authors present their careful designed solution to the problem.

**5.1 Content Adaptation**

Implementation depends on data from previous frame and can even use velocity data from previous frames as well. If some new object enter the frame it has to be calculated at a full rate. For performance benefit was used YUV value instead of full RGB channels.It has ben calculated twice for horizontal and vertical axis.For calculation is used multithread to ensure the fasted result, and is specific to hardware limitation.So it takes 0.2-0.3 ms from a frame time when calculated for a 4K image on GTX 2080 GPU.

**5.2 Motion Adaptation**

For frame velocity calculation can be use data from antialiasing pass of revise frame if is unavailable this data should been calculated again. Some static screen object should me marked for dissent approach because calculation depends on fast camera movement.

**5.3 Applying Adjusted Shading Rate**

At this state we use gathers data to apply selected VRS. Because VRS is a hardware feature it dos not require some addition steps to enable this so it greatly simplify integration on engine side.

**5.4 Optional Quality Improvement**

Sometimes shading rate can be unstable and oscillate between frames.To mitigate this issue we can use two thresholds fro different frame. Addition compute shader can be use to find such feedback loops. Also some material can be more sensitive to lowering rate of shading so implementation can use different threshold on per material bases.

# 6 RESULT

This technologie has been used in id Tech 6 and Unreal Engine 4.19. After implementation it has been tested to not show sings of image quality degradation. Motion adaptive shading as in game motion shader. Performance impact highly depends on content of a scene and camera angels. Average performance gain was 5-20% lower frame time on highly optimised games.Replacing standart motion blur shader with motion adaptive shading has provided even more benefits.

# 7 CONCLUSION AND FUTURE WORK

VRS can be used to improve performance of game and rendering engines. Authors hopes VRS will be highly adopted. For future work they want to include VRS in more shader pass and use it in ray-tracing passes.