Shading Atlas Streaming Demonstration

Joerg H. Mueller*
joerg.mueller@tugraz.at
Graz University of Technology

Thomas Neff*
thomas.neff@icg.tugraz.at
Graz University of Technology

Philip Voglreiter*
voglreiter@icg.tugraz.at
Graz University of Technology

Mina Makar
mmakar@qti.qualcomm.com
Qualcomm Technologies Inc.

Markus Steinberger
steinberger@icg.tugraz.at
Graz University of Technology

Dieter Schmalstieg
schmalstieg@tugraz.at
Graz University of Technology

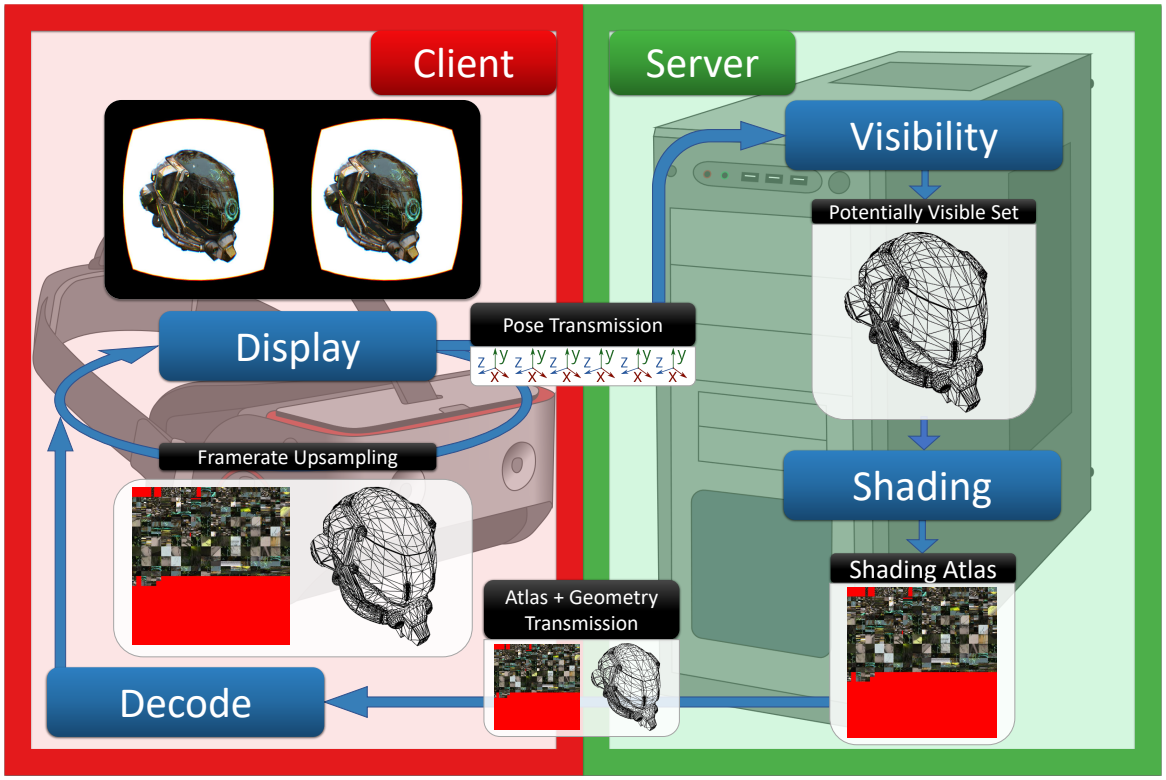

Figure 1: Our pipeline is split across a client and a server part, with the shading atlas as the central data structure connecting the two. Camera pose updates generated by the user are sent upstream, on a slow (networked) path to the server for rendering new shading into the atlas, and on a fast (direct) path to the client to render new images to the display. From the shading atlas and the geometry, novel views at close viewpoints can be rendered for framerate upsampling and warping. The shading atlas is temporally coherent and lends itself to efficient MPEG compression and streaming.

ABSTRACT

Streaming high quality rendering for virtual reality applications requires minimizing perceived latency. Shading Atlas Streaming (SAS) [Mueller et al. 2018] is a novel object-space rendering framework suitable for streaming virtual reality content. SAS decouples server-side shading from client-side rendering, allowing the client to perform framerate upsampling and latency compensation autonomously for short periods of time. The shading information created by the server in object space is temporally coherent and can be efficiently compressed using standard MPEG encoding. SAS compares favorably to previous methods for remote image-based rendering in terms of image quality and network bandwidth efficiency. SAS

*J. Mueller, T. Neff and P. Voglreiter are affiliated with the Christian Doppler Laboratory of Semantic 3D Vision established at Graz University of Technology.

SIGGRAPH '19 Emerging Technologies, July 28 - August 01, 2019, Los Angeles, CA, USA

ACM ISBN 978-1-4503-6308-2/19/07.
https://doi.org/10.1145/3305367.3327981

allows highly efficient parallel allocation in a virtualized-texture-like memory hierarchy, solving a common efficiency problem of object-space shading. With SAS, untethered virtual reality headsets can benefit from high quality rendering without paying in increased latency. Visitors will be able to try SAS by roaming the exhibit area wearing a Snapdragon 845 based headset connected via consumer WiFi.

CCS CONCEPTS

• **Computing methodologies** → **Rendering**; **Virtual reality**.

KEYWORDS

Streaming, Shading, Texture atlas, Object-space shading, Virtual reality

ACM Reference Format:
Joerg H. Mueller, Thomas Neff, Philip Voglreiter, Mina Makar, Markus Steinberger, and Dieter Schmalstieg. 2019. Shading Atlas Streaming Demonstration. In *Proceedings of SIGGRAPH '19 Emerging Technologies.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3305367.3327981

1 SYSTEM OVERVIEW

The SAS end-to-end pipeline is illustrated in Figure 1. The visibility stage determines which geometry is visible and needs to be shaded. This is done by computing a potentially visible set (PVS) of current and predicted viewpoints, using an extended field of view (FOV). We determine visibility on *patches* of two or three adjacent triangles. Atlas memory management is directly derived from the PVS, and is done dynamically on the GPU. Newly visible patches are allocated in the atlas; patches that have become invisible are removed from the atlas. Patch sizes are determined based on the projected area of a patch on the screen. In the shading stage, visible patches are shaded into the atlas, using a standard geometry pass. Since triangles do not overlap, no depth buffer is needed. Instead of screen space positions, the vertex shader outputs the shading atlas texture coordinates as position. We encode the updates to atlas and meta-information (geometry, patch and animation updates) before transmitting them to the client. During the visibility stage, we collect changes to the PVS data structures directly on the GPU, and therefore only require a single readback to the CPU in the encoding stage. While the network load is dominated by the atlas, it is temporally coherent, and can therefore be efficiently encoded on the GPU as an MPEG stream. We transmit the atlas as a lossy MPEG stream, relying on RTP over UDP. Since UDP is prone to packet loss, we insert I-frames at regular intervals to stop error propagation in the decoded atlas due to packet loss. We coordinate our memory management cleanup with the I-frames of the MPEG encoder, resulting in no loss of encoding efficiency. The meta-information is transmitted using TCP to ensure that the scene structure at the client is kept intact. The client decodes received messages immediately upon arrival and updates its internal structures. MPEG decoding on the mobile device is handled by a dedicated hardware unit. The meta-information is decoded on the GPU. Synchronization between MPEG and meta-information streams is done after each frame, and the decoding stage communicates with the display stage via triple buffering to avoid any latencies from locked buffers.

To display a frame, the client uses a simple forward rendering pass. The client issues a single draw call on a vertex buffer that has been generated by processing the incoming server data. This vertex buffer already contains the correct texture coordinates into the atlas, requiring no indirect lookup. Rendering is decoupled from the remaining pipeline; if an update has been received from the server and is ready to be used, the rendering thread simply switches buffers and continues rendering at full speed. Since the atlas contains shading information for an extended FOV as well as predicted frames, the client can use this information to render new poses between server updates, allowing for a smooth VR experience.

2 DEMONSTRATION

Our demonstration features a fully untethered end-to-end experience of SAS, using a Snapdragon 845 based headset as the client, capable of 6 DOF inside-out tracking. The client is connected via consumer WiFi to a server PC. Visitors will be able to move freely around our exhibit area, navigating around a virtual scene. Additionally, we will demonstrate a conventional streaming method, where raw eye-buffers are transmitted and warped on the client. Attendees will be able to switch between SAS and the conventional method at any time, allowing for a direct comparison. Visitors can also experience the effect of different network environments via a tethered headset, where network bandwidth and latency can be artificially modified. For bystanders, the view on the headset will be mirrored on a large screen connected to the server PC. Various visualization modes will visualize the inner workings of SAS, such as the atlas management or image quality comparisons to conventional game video streaming.

3 CONCLUSIONS

SAS demonstrates the first object-space shading approach for remote rendering with low perceived latency. By approximating a PVS, we can effectively hide latency on the client, as well as handle disocclusion events, while only using a lightweight client. Mipmapping is implicitly handled by the atlas memory allocation, and the temporally coherent memory management allows for very efficient MPEG encoding of the atlas. Compared to conventional streaming methods, SAS can handle the bandwidth requirements and visual fidelity of high-resolution content. For upcoming 5G networks, SAS can still benefit significantly from the increased network bandwidth, while enabling a responsive VR experience due to latency hiding. For detailed performance comparisons of all stages of SAS, as well as a discussion of our design decisions and possible extensions, please refer to our full paper [Mueller et al. 2018].

ACKNOWLEDGMENTS

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

REFERENCES

Joerg H. Mueller, Philip Voglreiter, Mark Dokter, Thomas Neff, Mina Makar, Markus Steinberger, and Dieter Schmalstieg. 2018. Shading Atlas Streaming. *ACM Transactions on Graphics* 37, 6, Article 199 (Nov. 2018), 16 pages. https://doi.org/10.1145/3272127.3275087