Adaptive Scalable Texture Compression

Adaptive Scalable Texture Compression (**ASTC**) is a lossy block-based texture compression algorithm developed by Jørn Nystad et al. of ARM Ltd. and AMD.[1]

Full details of ASTC were first presented publicly at the High Performance Graphics 2012 conference, in a paper by Olson et al. entitled "Adaptive Scalable Texture Compression"[2]

ASTC was adopted as an official extension for both OpenGL and OpenGL ES by the Khronos Group on 6 August 2012.[3]

Contents

Hardware support

Both profiles (LDR and Full) are supported on the latest Mali versions, including the Mali™-T620, Mali-T720, Mali-T760, Mali-T820/T830 and Mali-T860/T880.

Nvidia's Kepler and Maxwell-based Tegra SoCs.[4]

Intel GPUs in Skylake and later processors. [5]

On Linux, all Gallium 3D drivers have a software fallback since 2018, so ASTC can be used on any AMD Radeon GPU.[6]

Overview

The method of compression is an evolution of Color Cell Compression with features including numerous closely spaced fractional bit rates, multiple color formats, support for High Dynamic Range (HDR) textures, and real 3D texture support.

The stated primary design goal for ASTC is to enable content developers to have better control over the space/quality tradeoff inherent in any lossy compression scheme. With ASTC, the ratio between adjacent bit rates is of the order of 25%, making it less expensive to increase quality for a given texture.

Encoding different assets often requires different color formats. ASTC allows a wide choice of input formats, including luminance-only, luminance-alpha, RGB, RGBA, and modes optimized for surface normals. The designer can thus choose the optimal format without having to support multiple different compression schemes.

The choices of bit rate and color format do not constrain each other, so that it's possible to choose from a large number of combinations.

Despite this flexibility, ASTC achieves better peak signal-to-noise ratios than PVRTC, S3TC, and ETC2 when measured at 2 and 3.56 bits per texel.[2] For HDR textures, it produces results comparable to BC6H at 8 bits per texel.[2]

Example image prior to compression

Detail from example image, after compression at 8, 3.56 and 2 bits/pixel

Supported color formats

Encoding Format	Description
L	Luminance-only
LA	Luminance with transparency
L+A	Luminance with uncorrelated transparency
X+Y	Surface normals
RGB	Full color
XY+Z	Surface normals with uncorrelated Z
RGBA	Full color with transparency
RGB+A	Full color with uncorrelated transparency

Each of these may be encoded as low or high dynamic range. The encoder selects color formats independently for each block in the image.

2D block footprints and bit rates

ASTC textures are compressed using a fixed block size of 128 bits, but with a variable block footprint ranging from 4x4 texels up to 12x12 texels. The available bit rates thus range from 8 bits per texel down to 0.89 bits per texel, with fine steps in between.

Block footprint	Bit rate	Increment
4x4	8.00	25%
5x4	6.40	25%
5x5	5.12	20%
6x5	4.27	20%
6x6	3.56	14%
8x5	3.20	20%
8x6	2.67	5%
10x5	2.56	20%
10x6	2.13	7%
8x8	2.00	25%
10x8	1.60	25%
10x10	1.28	20%
12x10	1.07	20%
12x12	0.89	

In the above table, the "Increment" column shows the additional storage required to store a texture using this bit rate, as compared to the next smallest. Block footprints are presented as width x height.

3D block footprints and bit rates

ASTC 3D textures are compressed using a fixed block size of 128 bits, as for 2D but with a variable block footprint ranging from 3x3x3 texels up to 6x6x6 texels. The available bit rates thus range from 4.74 bits per texel down to 0.59 bits per texel, with fine steps in between.

Block footprint	Bit rate	Increment
3x3x3	4.74	33%
4x3x3	3.56	33%
4x4x3	2.67	33%
4x4x4	2.00	25%
5x4x4	1.60	25%
5x5x4	1.28	25%
5x5x5	1.02	20%
6x5x5	0.85	20%
6x6x5	0.71	20%
6x6x6	0.59	

Block footprints are presented as width x height x depth.

See also

- 3Dc

- Ericsson Texture Compression (ETC)
- PVRTC
- S3TC

References

1. "Adaptive Scalable Texture Compression (ASTC) technology developed by ARM® and AMD" (https://www.arm.com/products/multimedia/mali-technologies/adaptive-scalable-texture-compression.php).
2. "Adaptive Scalable Texture Compression" (https://www.cs.cmu.edu/afs/cs/academic/class/15869-f11/www/readings/nystad12_astc.pdf) (PDF). HPG 2012. Retrieved 2012-06-27.
3. "Khronos Releases ATSC Next-Generation Texture Compression Specification" (http://www.khronos.org/news/press/khronos-releases-atsc-next-generation-texture-compression-specification). The Khronos Group Inc. 2012-08-06. Retrieved 2012-08-06.
4. "Vulkan API" (http://on-demand.gputechconf.com/siggraph/2015/presentation/SIG1501-Piers-Daniell.pdf) (PDF).
5. "Graphics API Developer's Guide For 6th Generation Intel® Core™ Processors" (https://software.intel.com/en-us/articles/6th-gen-graphics-api-dev-guide).
6. https://lists.freedesktop.org/archives/mesa-dev/2018-July/200867.html

External links

- High Performance Graphics 2012 (http://www.highperformancegraphics.org/)
- ASTC Texture Compression: ARM Pushes the Envelope in Graphics Technology (http://community.arm.com/groups/arm-mali-graphics/blog/2011/12/13/astc-texture-compression-arm-pushes-the-envelope-in-graphics-technology/)
- ARM Unveils Details of ASTC Texture Compression at HPG Conference (http://community.arm.com/groups/arm-mali-graphics/blog/2012/06/28/arm-unveils-details-of-astc-texture-compression-at-hpg-conference--part-1/)
- ASTC Evaluation Codec from ARM (https://github.com/ARM-software/astc-encoder)
- Khronos ASTC Full Profile Extension Specification (http://www.khronos.org/registry/gles/extensions/OES/OES_texture_compression_astc.txt)
- UASTC (Universal ASTC) Texture Specification (https://github.com/BinomialLLC/basis_universal/wiki/UASTC-Texture-Specification)