Compact Alpha-texture Compression Unit for Mobile GPU

Jeongae Park, Won-Jong Lee, Sang Oak Woo, Seok-Yoon Jung, Shi-Hwa Lee

SAIT, SAMSUNG ELECTRONICS Co., Ltd.

SAMSUNG ADVANCED INSTITUTE OF TECHNOLOGY

Motivation

- Texture mapping is an essential technique in 3D graphics
- Issues
  - Texture size continually increases with the improvement in image quality
  - Texture mapping requires large memory area and memory bandwidth

- Solution: Texture compression
  - ETC1 [1] : adopted by OpenGL ES
    - Dose not support the alpha channel
  - S3TC [2] and ETC2 [3]
    - Support the alpha channel
    - Employ various compression methods over several modes
    - Require a large amount of hardware resources in implementations

- Proposal
  - Employ the RGB compression algorithm in ETC1 to also compress the alpha channel

References

[1] Jacob Strom and Tomas Akenine-Moller, "iPACKMAN: High-Quality, Low-Complexity Texture Compression for Mobile Phones," Graphics Hardware 2005, pp.63-70, 2005.

[2] Konstantine I. Iourcha, Krishna S. Nayak, Zhou Hong, “System and method for fixed-rate block-based image compression with inferred pixel values192,” US Patent 5956431.

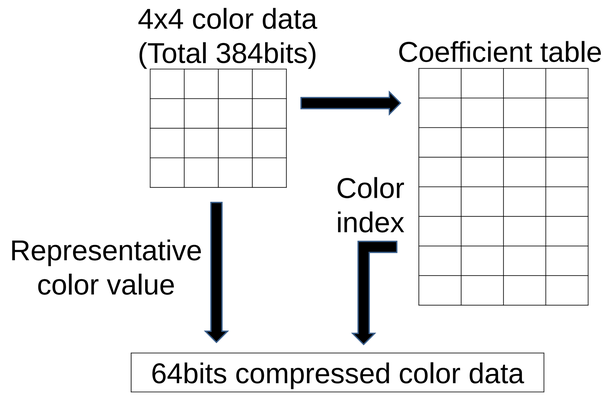
[3] Jacob Ström and Martin Pettersson, “ETC2: Texture Compression using Invalid Combinations,” Graphics Hardware 2007, pp.49-54, 2007.

[4] Ericsson, etcpack, EricssonDeveloperTools, http://devtools.ericsson.com/

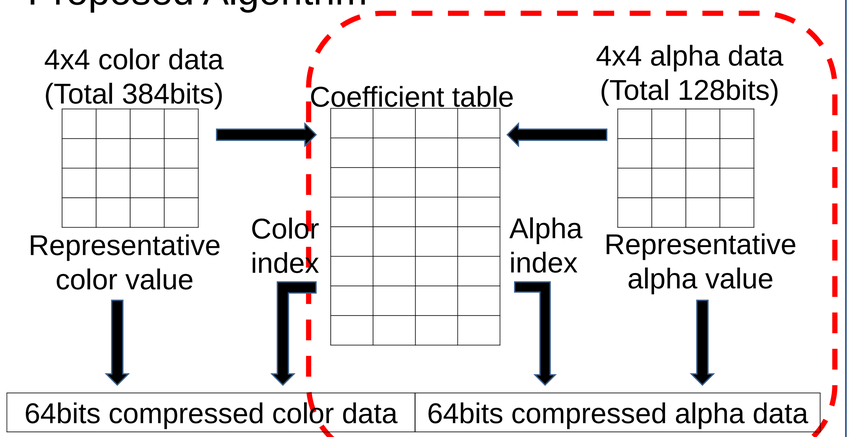
[5] Rightware, Basemark ES 2.0, http://www.rightware.com/en/Benchmarking+Software/Basemark%99++ES+2.0/

Proposed Method

✓ ETC1 Algorithm



✓ Proposed Algorithm



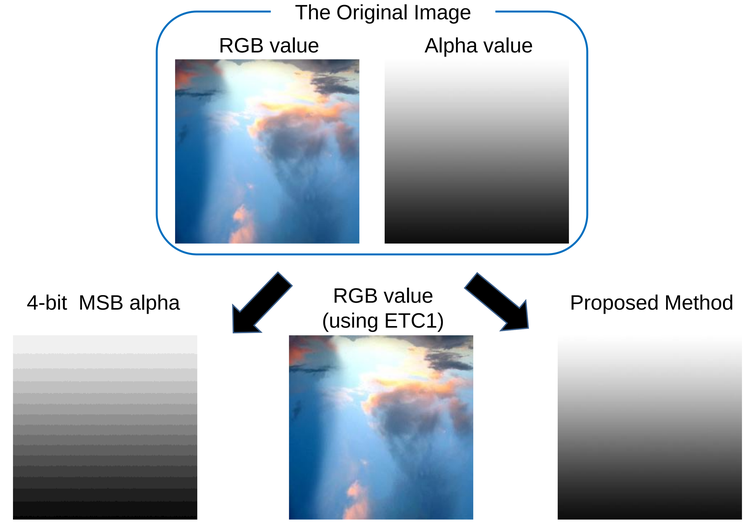
- Allocate 8-bits for an alpha value
  - Because the loss in the alpha channel lowers the image quality more severely (cf. 4~5-bits for R/G/B)

- Fixed-ratio (4:1) compression
  - Allow random access to any pixel

- Generate index values for the alpha channel using the same coefficient table in ETC1
  - Minimize the increase in the hardware cost by reusing the table that is used for RGB compression

Experiments

✓ Proposed Method vs. 4-bit MSB alpha



✓ Compressed Data Size

	8-bit alpha	4-bit MSB alpha	Proposed method
Amount of data (Byte/Block)	24	16	16
Compression ratio	1	0.67	0.67

✓ PSNR (dB)

	4-bit MSB alpha (A)	Proposed method (B)	Improvement (B-A)
Sky	29.3	51.4	22.1
Wall	30.0	42.3	12.3
Gravel	29.7	39.1	9.4
Tree	29.7	38.6	8.9
Lattice	30.8	38.5	7.7
Rampart	30.3	35.9	5.6
Pattern	31.7	36.4	4.7

- Experimental setup
  Ericsson etcpack [4] + Basemark ES 2.0 [5] Taiji
- Reduce the texture memory size to 66.7%, as compared to the non-compressed alpha data
- Increase the PSNR by 4.7~22.1dB, as compared to 4-bit MSB alpha values, which requires the same amount of texture memory

SAIT