INTRODUCTION

- HTJ2K, file extension .jph, is the latest addition to JPEG2000 Suite of image coding standards. It is also known as JPEG2000 Part 15, ISO/IEC 15444-15, and ITU-T.H.814.
- The most demanding step in conventional JPEG2000 (J2K) is the block coder – visiting data multiple times, and the serial nature of the context adaptive arithmetic coder.
- HTJ2K describes a “fast” block coder – codes many blocks at once and is highly parallelizable.
- Retains J2K-1 features, capabilities, and is compatible (losslessly-tocodable) with J2K-1 – supports limited quality scalability.
- Block coding speedup of ~10x (lossy) ~40x (lossless).
- Slightly lower coding efficiency compared to J2K.
- HTJ2K describes a “fast” block coder – codes many blocks at once and is highly parallelizable.
- HTJ2K supports color transform at the kernel level (CUP).
- Kakadu JPEG2000 SDK v8.0 supports it.
- Slightly lower coding efficiency compared to J2K.

The HTJ2K BLOCK CODESTREAM

- The codestream comprises up to 3 coding passes: a Cleanup Pass (CUP), followed by an optional Propagation Pass (SPP), and a yet optional Magnitude Refinement Pass (MRP).
- The CUP has three byte-streams: MagSgn (forward), MEL (forward), and VLC (backward).
- Forward-backward exposes more parallelism. Here, we decode MEL and VLC, and SPP together. Later, we decode MagSgn, add SPP results, and decode MRP at the same time.

A GPU-BASED DECODER FOR HTJ2K

The CPU parses a JPH file, generating lists of code-block information (data locations, segment sizes, … etc.), which are transferred to GPU.

- In this work, the GPU operates in one of two scenarios:
  - Kernels with No Refinements (NR) – decoding CUP Only
  - A practical decoder can always discard SPP, MRP – lower quality
  - KCUPS1 (serial per code-block): decodes MEL & VLC byte-streams, stores decoded info (*w, *n, *x, *y) in global memory. VLC tables are transferred to shared memory by first warp. Uses one thread per codeblock, and 45 registers.
  - KCUPS2 (parallel per codeblock): retrieves (*w, *n, *x, *y) from global memory and decodes MagSgn byte-stream, generating decoded coefficients. Employs one wrap per 64x64 codeblock, and 64 registers.
  - WSYN: performs wavelet synthesis on all resolution except the last. Also receives info from CPU about all-zero code-blocks in order to skip data retrieval for these blocks.
  - WSYN+Color: similar to the above, but performs color transform at the end. It stores the data ready for transfer to CPU. Uses 125 registers.
  - All processing is performed using 32 bit floats.
- Kernel with Refinements (R) – decoding CUP, SPP, MRP.
  - KCUPS1-SPP: similar to KCUPS1, but also decodes SPP, for which it stores 2bits/sample in global memory. Uses 77 registers and 144 bytes shared memory as a scratchpad.
  - KCUPS2+MRP: similar to KCUPS2, but also decodes MRP. It also retrieves and combines decoded SPP information. Uses 82 registers.

EXPERIMENTAL RESULTS

- Results are for 4K 4:4:4 12bit video test sequence ARRi AlexaDrums.
- 64x64 code-blocks, irreversible CDF97 wavelet, 5 levels of decamps
- No overlap in frame decoding is employed.
- Compressed image are uploaded while earlier frames are decoded.
- Frame decode rates are obtained decoding 1000 HTJ2K frames.
- 3 GPU are test: low-end GT1030 with GDDR5, mid-range GTX 1060, and “enthusiast” GTX1080.

CONCLUSIONS

Decoding HTJ2K files on a GPU is feasible and can achieve very high frame rates, even on low-end GPUs; it is many folds faster than JPEG2000. Decoding 8K 4:4:4 HDR at 120 fps is possible on a GTX1080. Next, we will explore GPU encoding.