SIPL Activity on High Efficiency Video Coding

Yair Moshe

SIPL annual event – July, 2015
Video Coding Standards

- ITU-T Standards
  - H.261
  - H.263
  - H.263+
  - H.263++

- Joint ITU-T/MPEG Standards
  - H.262/MPEG-2
  - H.264/AVC

- MPEG Standards
  - MPEG-1
  - MPEG-4

Timeline:
- 1984
- 1986
- 1988
- 1990
- 1992
- 1994
- 1996
- 1998
- 2000
- 2002
- 2004
- 2006
- 2008
- 2010
- 2012
HEVC Milestones

- January 2010: Call for Proposals
- February 2012: Committee draft
- July 2012: Draft international standard
- January 2013: Final draft international standard
- April 2013: Approved as ITU standard
- July 2013: Formally published by ITU
- November 2013: Formally published by ISO/IEC
- October 2014: v2 approved as ITU standard (range extension, scalability, multi-view)
Google Trends

![Graph showing trends for mpeg-2, h.264, and hevc over time]

- mpeg-2
- h.264
- hevc
HEVC Main Features

Source formats
- Resolution: (320x240, 15 fps) – Ultra HDTV (8192x4320, 120 fps)
- Bit depth: 8 to 16 bpp
- Chroma format: 4:2:0, 4:2:2, 4:4:4
- Progressive scanning

Coding efficiency & complexity
- 30%-60% bit rate reduction at same quality compared with H.264 High profile
- Complexity-scalable: ½ to 3 times of H.264 High profile
- Compression ratio up to 1000:1

Designed for efficient parallel processing
Performance

x264
Tears Of Steel
1080P @ 500 Kbps

x265
Tears Of Steel
1080P @ 500 Kbps
HEVC Basics

- Designed from scratch
- Same hybrid predictive/transform coding structure as previous standards
- Many incremental improvements
HEVC has more flexibility than previous standards in partitioning each frame into variable sized structures. This is the most significant improvement:

- In terms of structure change
- In terms of rate-distortion
The basic working unit is called **Coding Tree Unit (CTU)**
- Analogous to macroblock in earlier standards
- 16x16, 32x32 or 64x64 pixels in size
- Fixed per video sequence

Each CTU can be divided into **Coding Units (CUs)**
- Square regions using a quadtree structure
- As small as 8x8 pixels
- Inter/intra decision is made at this level

Each CU is further partitioned in **Prediction Units (PUs)** and **Transform Units (TUs)**
Partitioning
Invented in 1974

Investigated for image compression in the past

Low Bit-Rate Video Coding Using Iterative Affine Motion Estimation and Quadtree Segmentation

O. Reshef and D. Malah

International Conference on Digital Signal Processing (DSP95), Limasol, Cyprus, June 1995.

Used in MPEG-4 for shape coding
Partitioning

H.264

HEVC
SIPL Activity on HEVC

- A tool for analyzing quatree partitioning techniques
- Fast HEVC partitioning algorithm
- Fast HEVC partitioning algorithm designed for GPU parallelization
- Low bitrate underwater video compression
A graphic MATLAB tool for investigating different partitioning criteria
For still images
5 different criteria, a lot of parameters to play with
Consider adjacent blocks depth partition
- Both spatial and temporal neighbors
- Discard certain depths

Early termination based on intermediate computations
Use heuristics
A technique for fast partitioning using adjacent blocks depth partition
Uses some clever heuristics
Implemented in the HEVC committee encoder
Tested with standard videos
Results compared with committee code (Choi & Jang, 2012)
- About 36% reduction in running time
- About 2% increase in bitrate

Results compared with state-of-the-art (Fan et al., 2014)
- About 5% reduction in running time
- Negligible difference in bitrate
Video encoding incurs high computational complexity

GPUs (Graphics Processing Units) have massive computation power
  - Very efficient in executing highly parallel tasks

It is desirable to use the computation power of GPUs for HEVC encoding
The problem: High dependency between neighboring CTUs

Few recent works try to use GPU for HEVC encoding
- Parallelization is inside each CTU
- Fast motion estimation and mode decision
- Suboptimal use of GPU due very limited parallelization
A technique for fast partitioning with reduced dependencies

Each CTU is encoded independently
  - Allows many threads to execute in parallel
  - Use only information from previous frames
Is this going to hurt compression efficiency?

**Surprise:** No!

Results compared with state-of-the-art (Fan et al., 2014)

- About 0.5% **decrease** in bitrate for low resolutions
- About 4% **decrease** in bitrate for high resolutions
Compression of 640x480 underwater videos

x264 @ 50 Kbps

x265 @ 30 Kbps
Much better compression for HEVC in spite of low resolution

- Large smooth areas
- Slow motion
HEVC is the future/current state-of-the-art video coding standard

Has some important advantages compared with previous video compression standards

There are many interesting things to do with HEVC

SIPL is currently doing some of them 😊
THANK YOU