JOANNEUM RESEARCH DIGITAL

DIGITAL - Institute for Information and Communication Technologies

GPU-Accelerated Undecimated Wavelet Transform for Film and Video Denoising

Hermann Fürntratt, Hannes Fassold

2015-03-19

SEVENTH FRAMEWORK PROGRAMME

DAVID

Overview

- GPU-activities @ AVM research group
- Undecimated Wavelet Transform
  - Algorithm
  - GPU Implementation
- Application example
  - Film and Video denoising

Overview

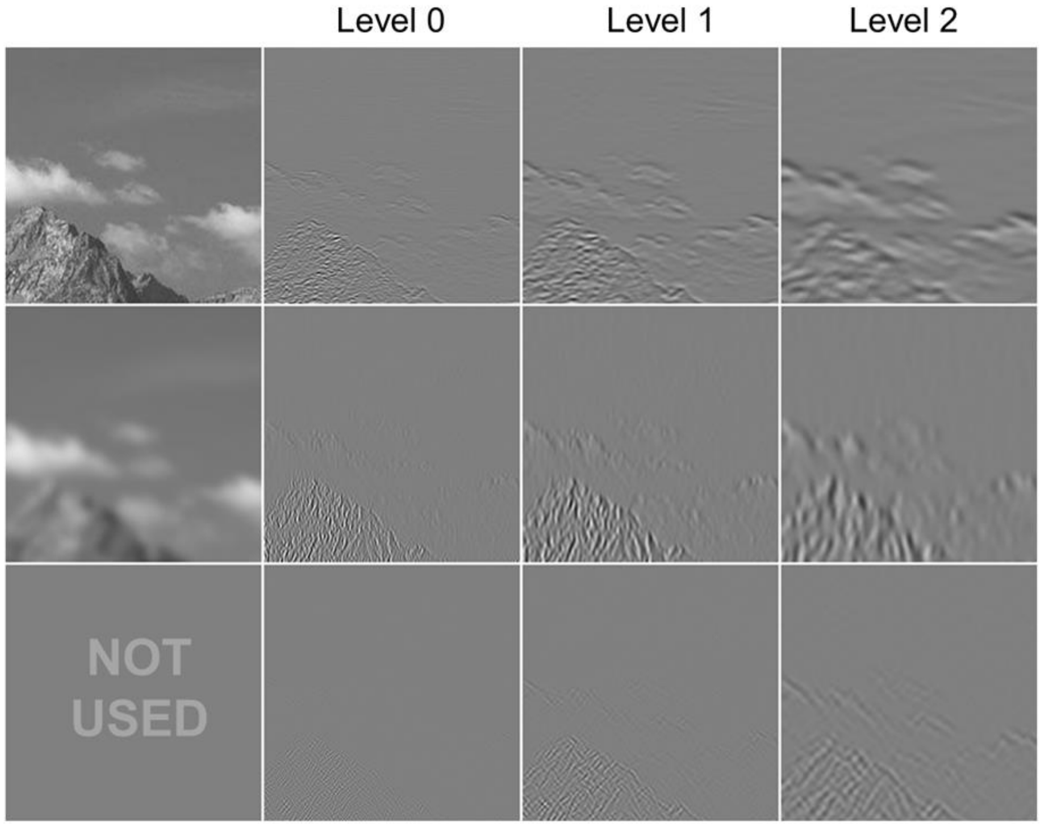
- GPU-accelerated algorithms / applications @ AVM
  - AVM - AudioVisual Media research group, DIGITAL – Institute for Information and Communication Technologies, JOANNEUM RESEARCH
  - Content-based video quality analysis http://vidicert.com
  - Digital film restoration http://www.hs-art.com
  - Brand monitoring http://www.branddetector.at
  - GPU activities since 2007 - using CUDA C++
  - Sucessfully ported complex computer vision algorithms like **KLT feature point tracking, SIFT descriptor extraction or Semi-Global Matching** to the GPU

Undecimated Wavelet Transform (1)

- Discrete Wavelet Transform (DWT) widely used
  - E.g. in image compression (JPEG 2000)
- Undecimated Wavelet Transform (UWT)
  - Also known as Stationary Wavelet Transform, Shift-Invariant Discrete Wavelet Transform, Overcomplete Discrete Wavelet Transform, see [Fowler2005] …
  - UWT is nearly the same as DWT, but the **sub-sampling step is skipped**
    - All wavelet components in all levels have the *same* size as the input image
    - Three wavelet components (LH, HL, HH) per level
  - Much better suited than DWT for all sort of image enhancement tasks
    - Denoising, deblurring, superresolution, …
  - Main disadvantage is the significantly higher computational complexity

Undecimated Wavelet Transform (2)

- UWT implementation
  - Calculated with ‚a trous‘ algorithm
  - Key routines used in each level
    - Convolution with **two 1-D row or 1-D column kernels** (one input image, two output images)
    - Convolution with **one 1-D row or 1-D column kernel**
  - Convolution kernel
    - Kernel size is growing for each level
    - Convolution kernel is getting progressively more ‚sparse‘



Undecimated Wavelet Transform GPU Implementation (1)

Goals

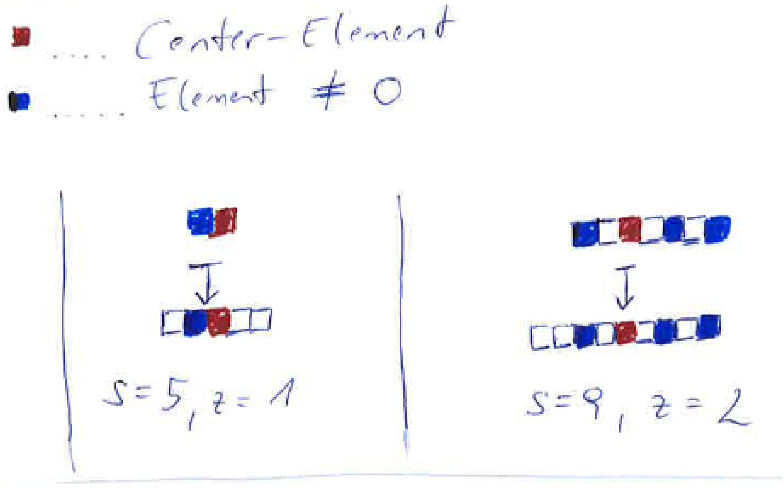
- Flexibility
  - Support for different wavelet classes (Haar wavelets, Daubechies wavelets, …) and different datatypes (16-bit float, 32-bit float)
- Maintainability
- Performance

Design principles of GPU implemention

- Loosely based on principles mentioned in [Iandola2013]
- **Load** directly into register **via ‚texture path‘**
- Computation of **multiple outputs per thread** (parameter ‚grainsize‘)
- Make it easy for compiler to **unroll the innermost convolution loop** by hard-coding loop bounds & loop increment

Undecimated Wavelet Transform (2)

- GPU implementation
  - Templatized CUDA kernel
    - Template parameters: datatype, convolution kernel radius, loop increment, grainsize
  - Algorithm workflow
    - For certain kernel radii and loop increments, the templatized CUDA kernel is called
      - A big **switch** clause with multiple *case* statements
    - Convolution kernels which are not symmetric (e.g. Haar wavelets) are **extended** to the next bigger symmetric convolution kernel for which a kernel call is **available** within the *case* statement



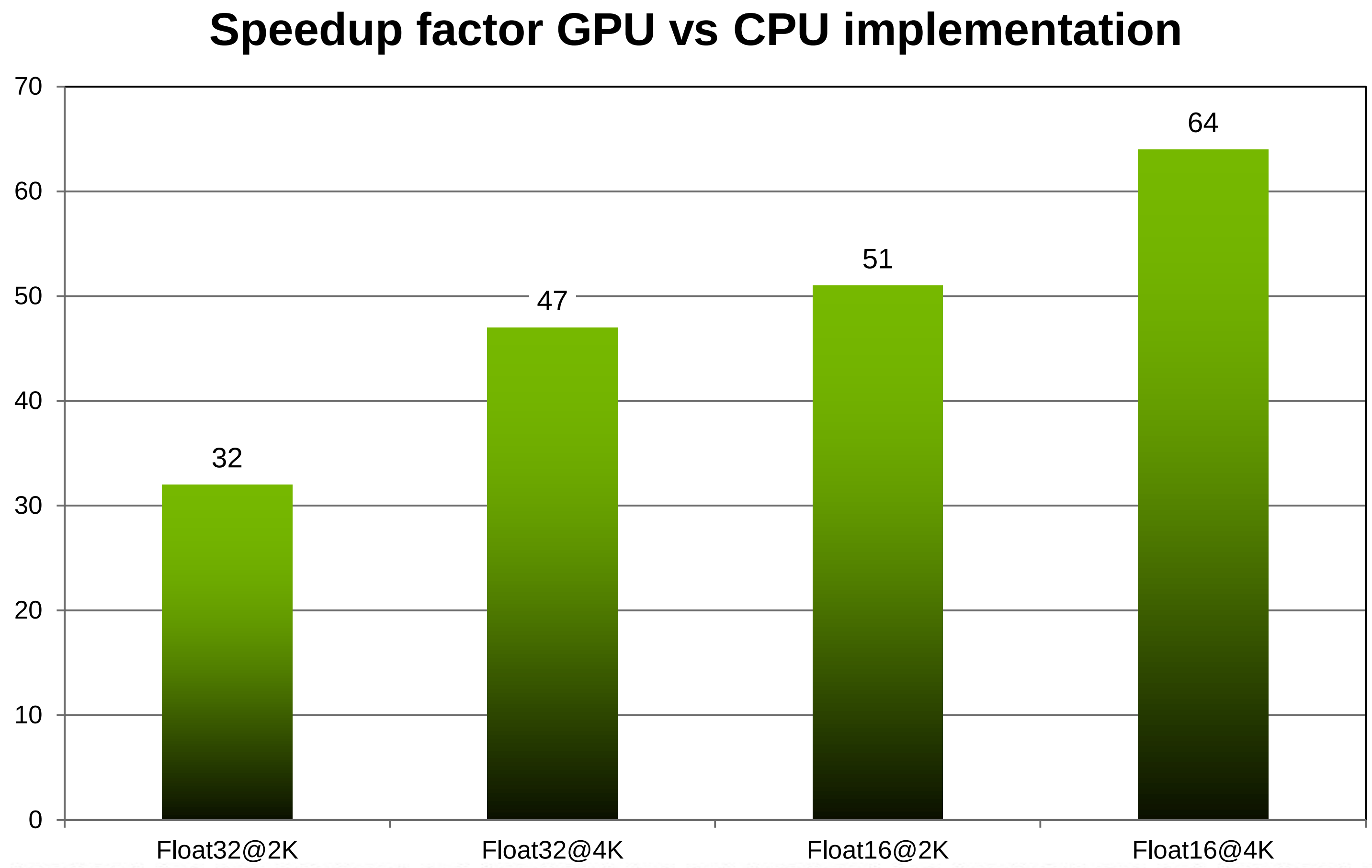
Undecimated Wavelet Transform (3)

- GPU implementation – some more notes
  - Constant memory is very useful
  - Texture path very useful
    - Makes CUDA kernels much more straightforward
      (no explicit handling of border pixels necessary, …)
    - Texture objects are very convenient (CC >= 3.0 necessary)
    - Good performance (good caching behavior) also for pitch-linear 2-D memory
  - Unrolling of innermost convolution kernel is very important
    - Disabling it makes CUDA kernel **two times slower**
  - Usage of 16-bit floats increases performance
    ~ 30 - 40%

Runtime comparison (1)

- CPU implementation
  - In-house C++ reference implementation
  - Multi-threaded using OpenMP
- Quality tests
  - Differences between GPU and CPU implementation are negligible
- Test setup for runtime tests
  - CPU: Xeon E5-1620 Quad-Core @ 3.6 GHz
  - GPU: NVIDIA GeForce GTX 770
  - Transfer time (CPU – GPU memory) not included

Runtime comparison (2)



2K: 2048 x 1556
4K: 4096 x 3112

GPU: GTX 770
1536 cores
@ 1.046GHz
CPU: Xeon E5-1620
4 cores
@ 3.6GHz

Film and video denoising (1)

- Noise
  - Can be often observed in both film (as film grain) and digital video (as digital sensor noise)
  - Degrades viewing experience considerably
  - Lowers the compression ratio when encoding noisy content

Noisy versions of ‚Lena' showing fine noise and coarse noise

Film and video denoising (2)

- Typical workflow of a wavelet-based denoising algorithm
  - Apply wavelet transform to one image or a (usually motion-compensated) 3-D spatiotemporal volume
  - Shrink insignificant (small-magnitude) wavelet coefficients towards zero
  - Apply inverse wavelet transform
- A practically usable denoising algorithm includes much more steps
  - Must be able to estimate the noise type (magnitude, coarseness, signal-dependency) automatically
  - Must have safeguards against motion-compensation errors

Film and video denoising (3)

- Novel denoising algorithm developed at AVM group
  - Automatically estimates noise characteristics
    - Is therefore able to adapt to the actual noise type (film grain, digital sensor noise, …)
  - Two-phase approach
  - Uses motion-compensated 3-D volume
- Evaluation results
  - Generated novel dataset where realistic noise was added with texture synthesis method
  - Evaluation shows that algorithm is competitive with best algorithm from academics CV-BM3D [Dabov2007]
- **< DEMO VIDEO (SPLIT-SCREEN) FOLLOWS >**

References

- [Dabov2007] K. Dabov, A. Foi, K. Egiazarian, “Video denoising by sparse 3d transform-domain collaborative ltering”, Proc. 15th European Signal Processing Conference (EUSIPCO), Poznan, Poland, 2007
  - Matlab implementation of CV-BM3D available for non-profit scientific research purposes at http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_software
- [Fowler2005] J. Fowler, „The Redundant Discrete Wavelet Transform and Additive Noise”, IEEE Signal Processing Letters, Volume 12, 2005
- [Iandola2013] F. Iandola, D. Sheffield, M. Anderoson, P. Phothilimhana, K. Kreutzer, „Communication-minimizing 2D convolution in registers“, Proc. IEEE International Conference on Image Processing, Melbourne, Australia, 2013.

Acknowledgments

- Thanks to the Flemish Radio and Television Broadcasting Organization (VRT, Belgium) for providing the video "elfenheuvel" for research and demonstration purposes.
- Thanks to the BM3D authors (K. Dabov, A. Foi, K. Egiazarian, Tampere University of Technology) for making available the Matlab code of the CV-BM3D algorithm for non-profit scientific research purposes.
- The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600827, DAVID (“Digital AV Media Damage Prevention and Repair").
  - http://david-preservation.eu/

SEVENTH FRAMEWORK PROGRAMME

DAVID

Thank you for your attention

Hermann Fürntratt, Hannes Fassold

hermann.fuerntratt@joanneum.at

hannes.fassold@joanneum.at

JOANNEUM RESEARCH
Forschungsgesellschaft mbH

Institute for Information and
Communication Technologies

www.joanneum.at/digital