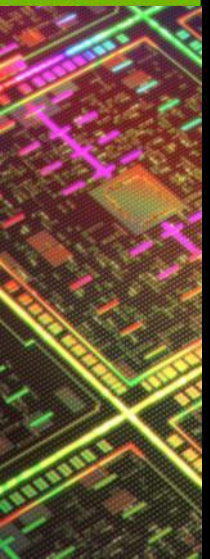
Fast JPEG Coding on the GPU

Fyodor Serzhenko, Fastvideo, Dubna, Russia
Victor Podlozhnyuk, NVIDIA, Santa Clara, CA



Key Points

- We implemented the fastest JPEG codec
- Many applications using JPEG can benefit from our codec

High Speed Imaging

Data Path for High Speed Camera (500 – 1000 fps)

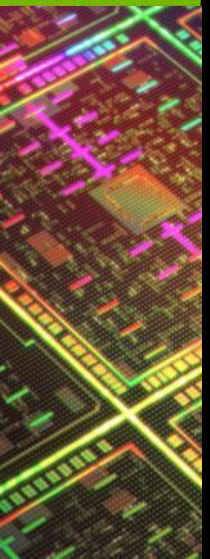
Camera data rate from 600 MB/s to 2400 MB/s.

Problem: how to record 1 hour or more?

Possible Solutions

RAID, SSD, online compression on FPGA / DSP / CPU / GPU

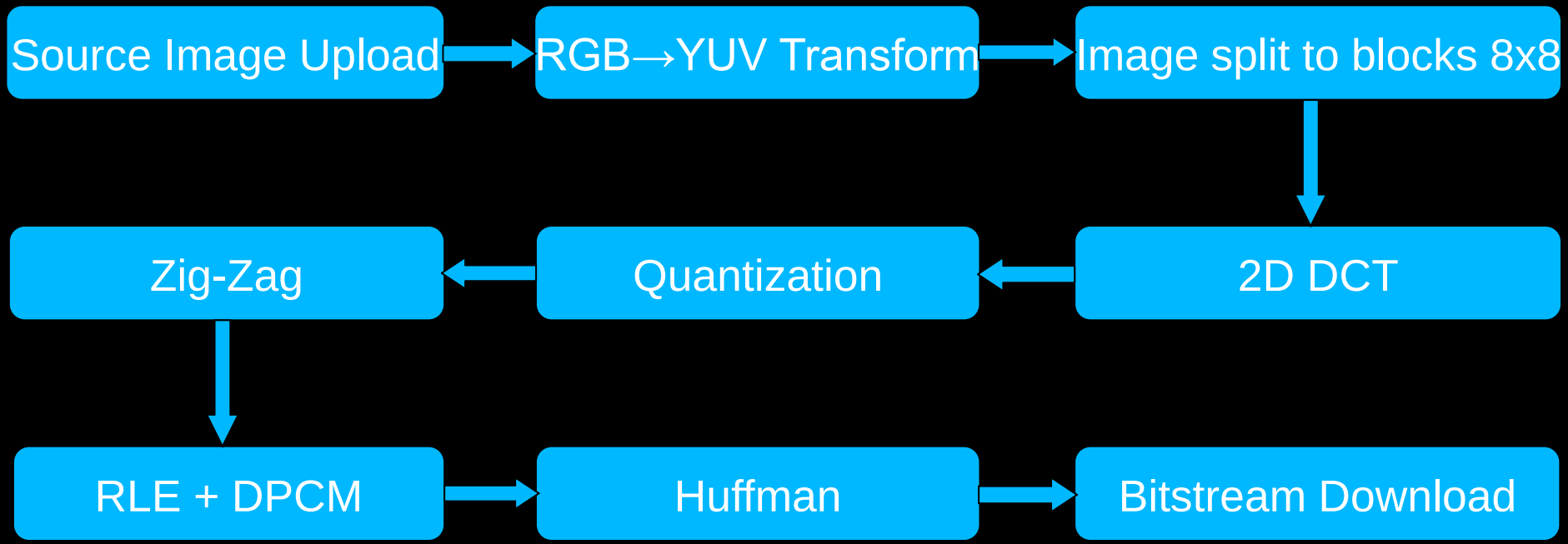
The fastest solution: JPEG compression on GPU

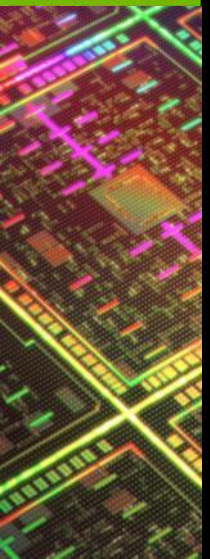


Why JPEG

- Popular open compression standard
- Good image quality at 10x-20x compression ratio
- Moderate computational complexity

Main Stages of Baseline JPEG Algorithm





JPEG Codecs: GPU vs. CPU

Performance summary for the fastest JPEG codecs

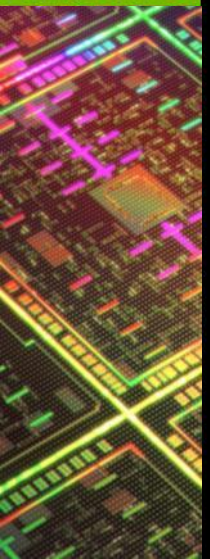
JPEG Codec (Q=50%, CR=13)	Encode, MB/s	Decode, MB/s
Fastvideo FVJPEG + GTX 680	5200	4500
Fastvideo FVJPEG + GTX 580	3500	3500
Intel IPP-7.0 + Core i7 3770	680	850
Intel IPP-7.0 + Core i7 920	430	600
Vision Experts VXJPG 1.4 (*)	500	--
Accusoft PICTools Photo (*)	250	380

(*) - as reported by manufacturer

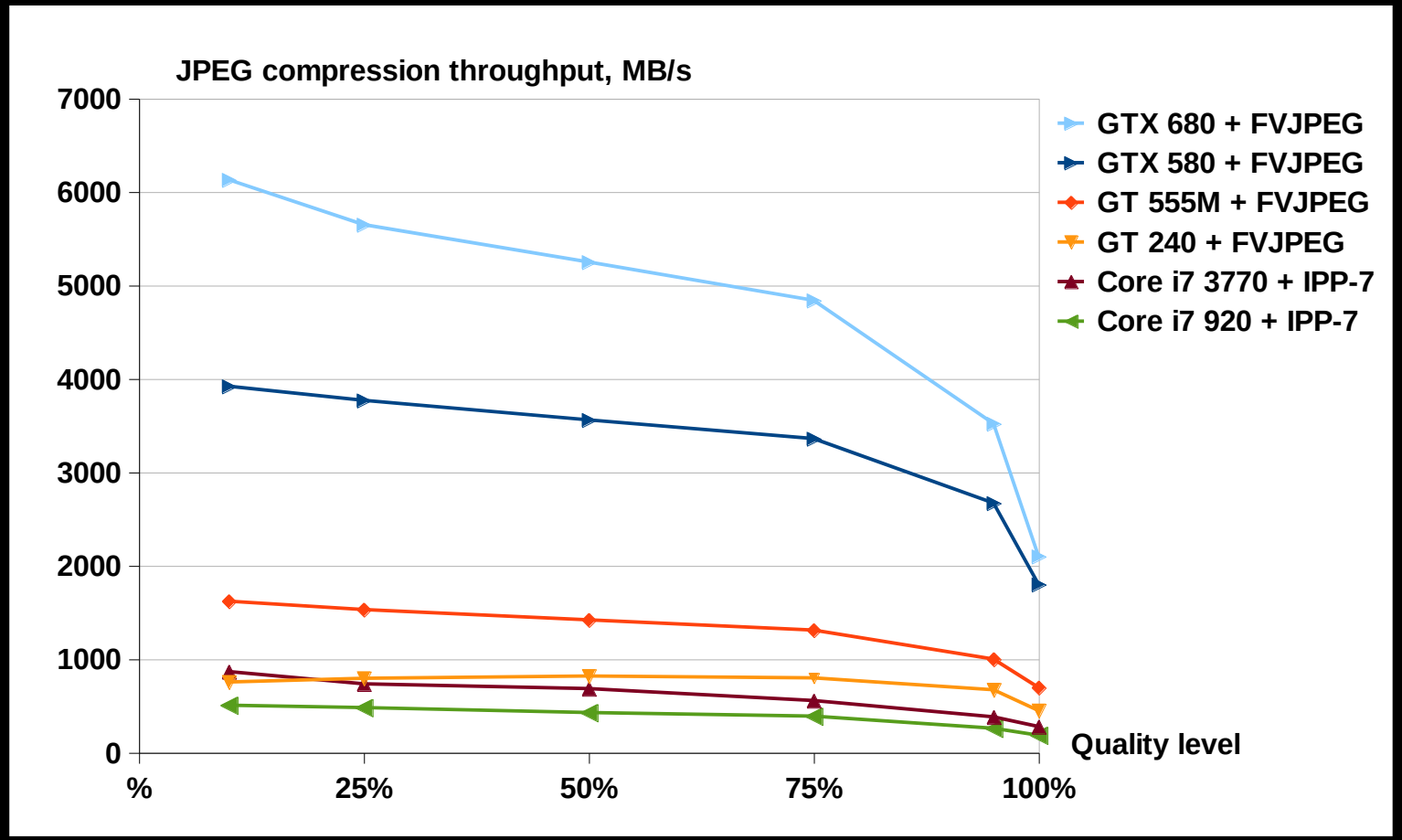
Best JPEG encoder IP Cores

JPEG IP Core	Encode MB/s
Cast Inc. JPEG-E	750
Alma-Tech SVE-JPEG-E	500
Visengi JPEG Encoder	405

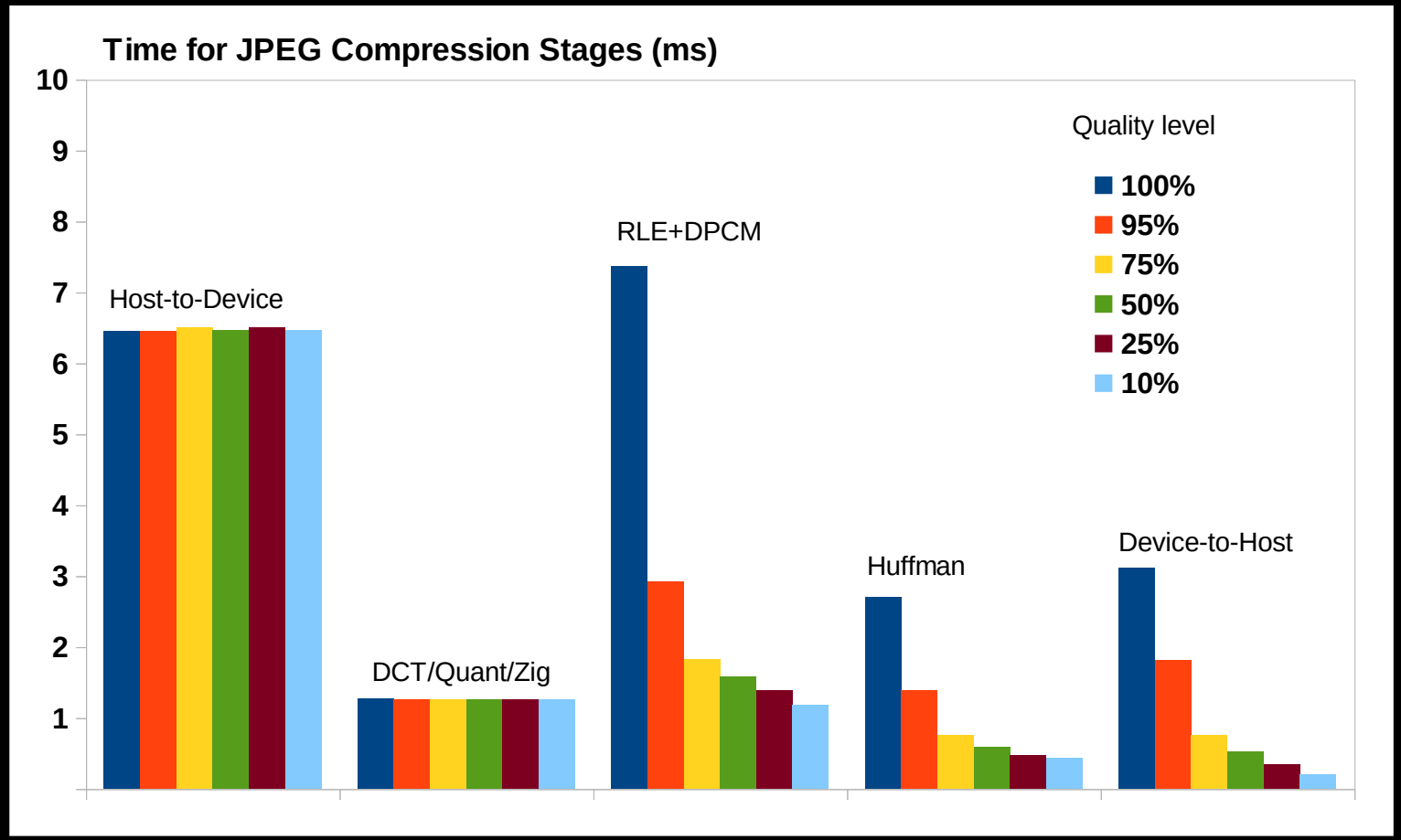
Results as reported by manufacturer



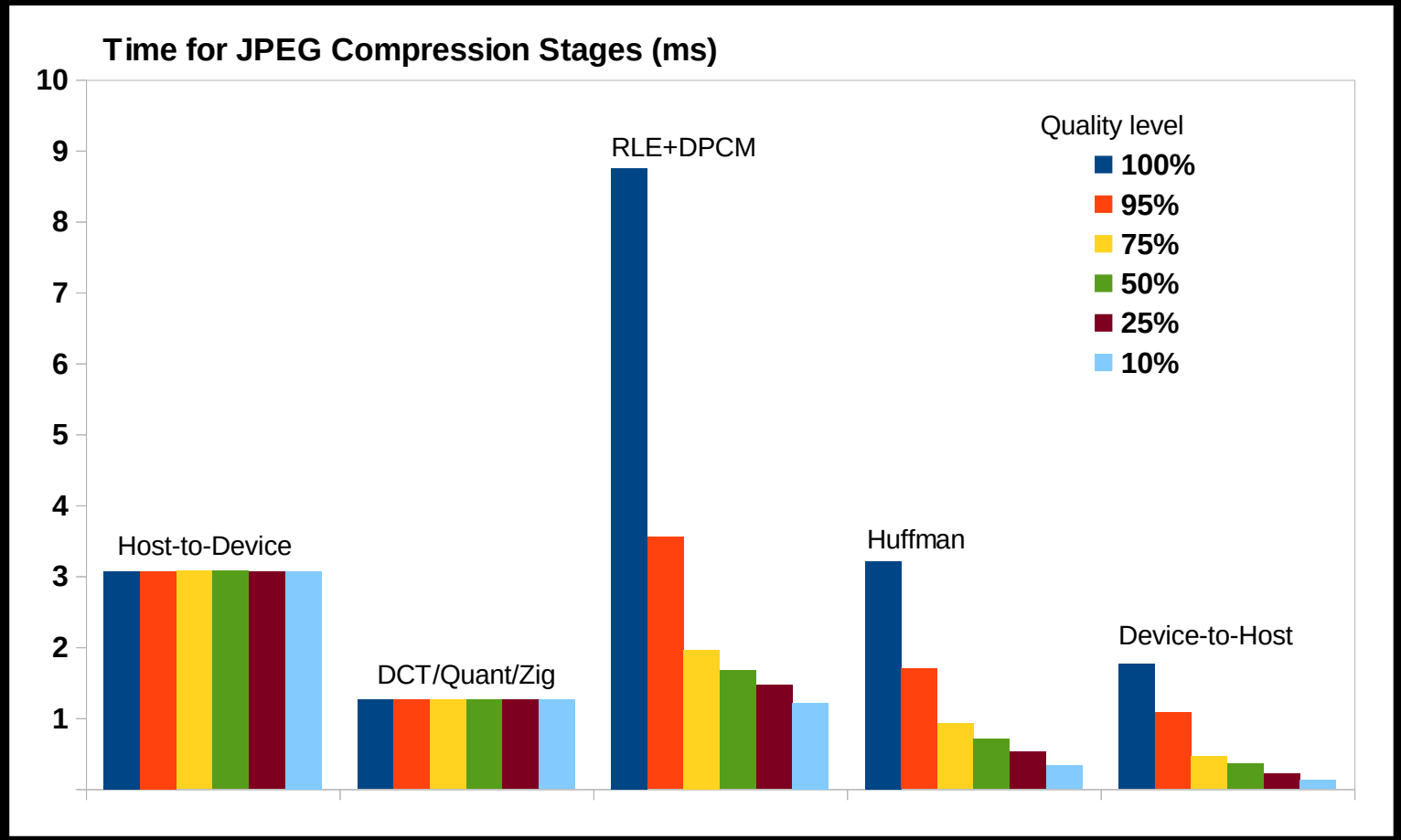
JPEG Encoding Rates for GPU & CPU



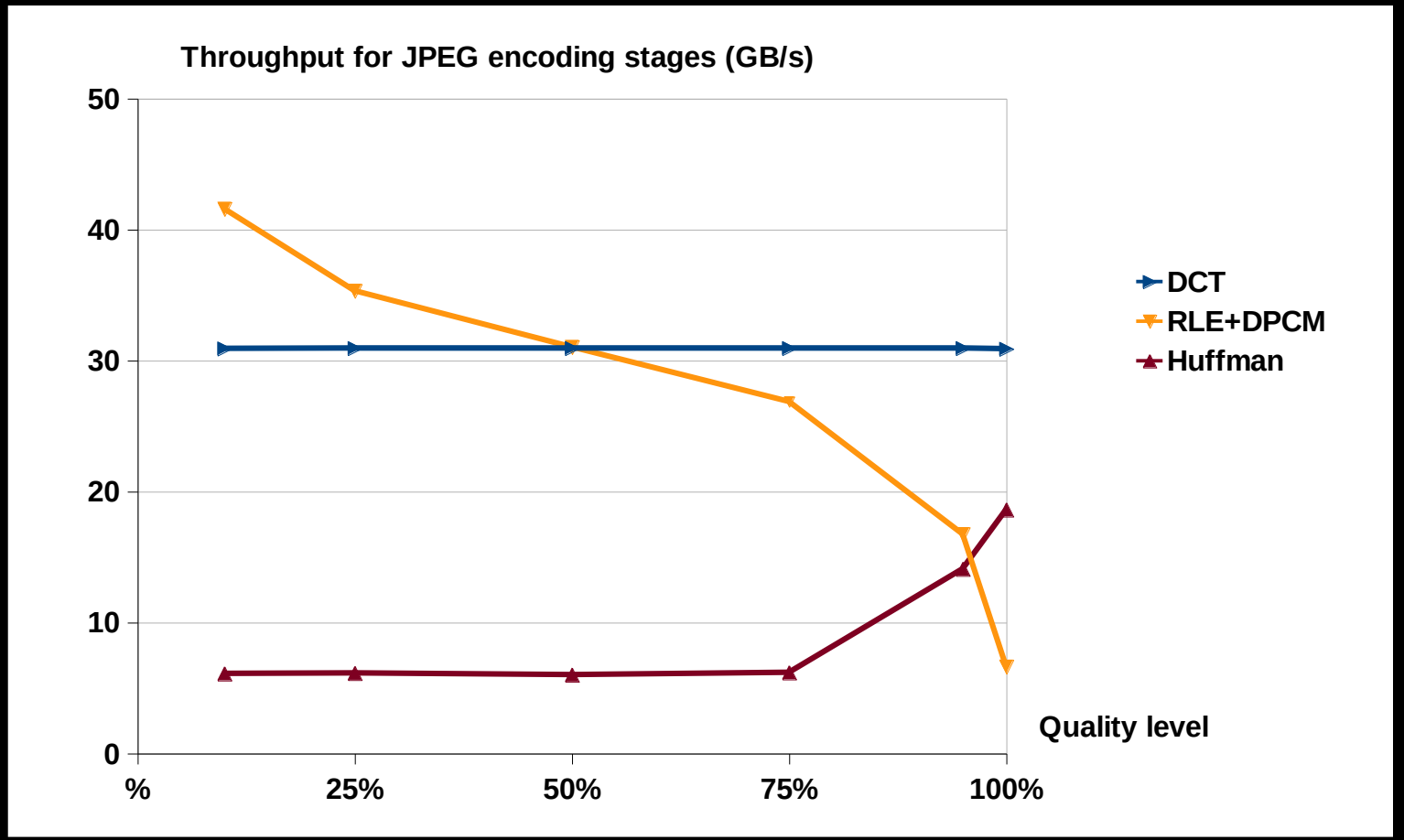
JPEG Encoding Time for GeForce 580

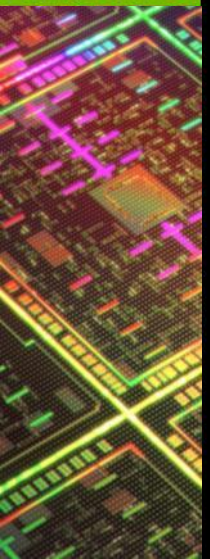


JPEG Encoding Time for GeForce 680



DCT and Entropy Encoding (GeForce 580)

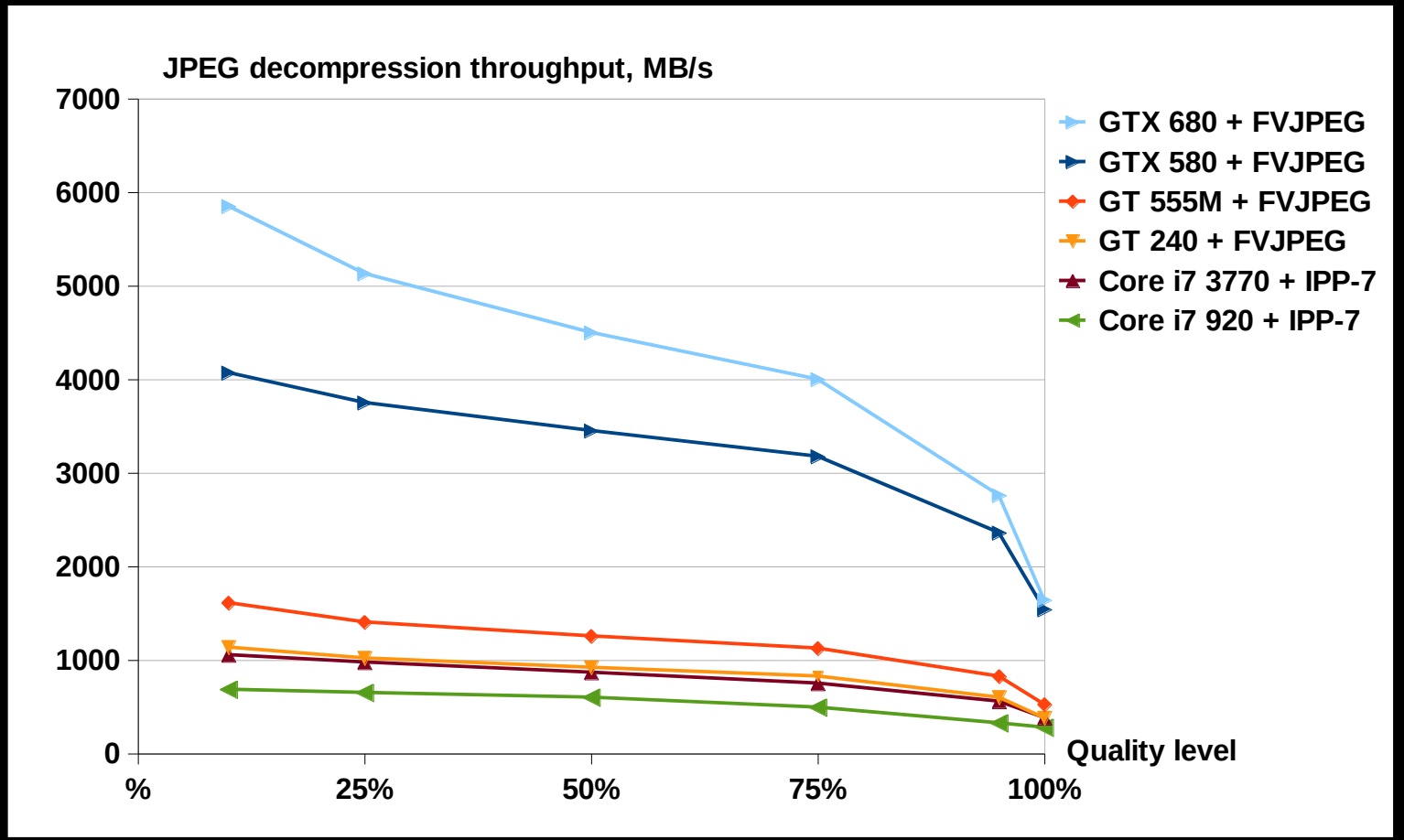




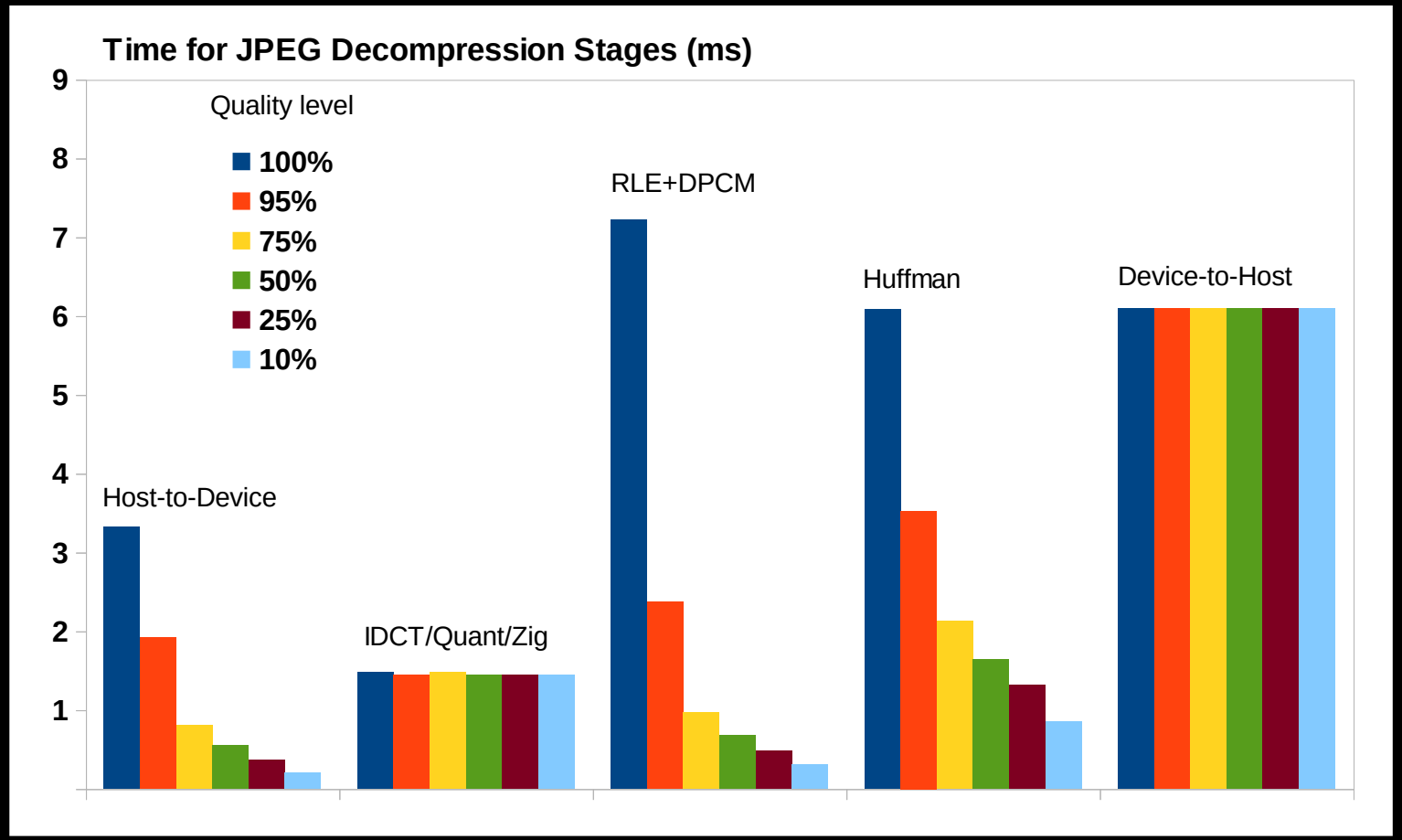
JPEG Decoding

- No good parallel algorithm is known for Huffman decoding
- Restart markers is a standard feature supported by all decoders
- Fully parallel JPEG decoding is still possible
- Currently supported restart intervals: 0, 1, 2, 4, 8, 16, 32

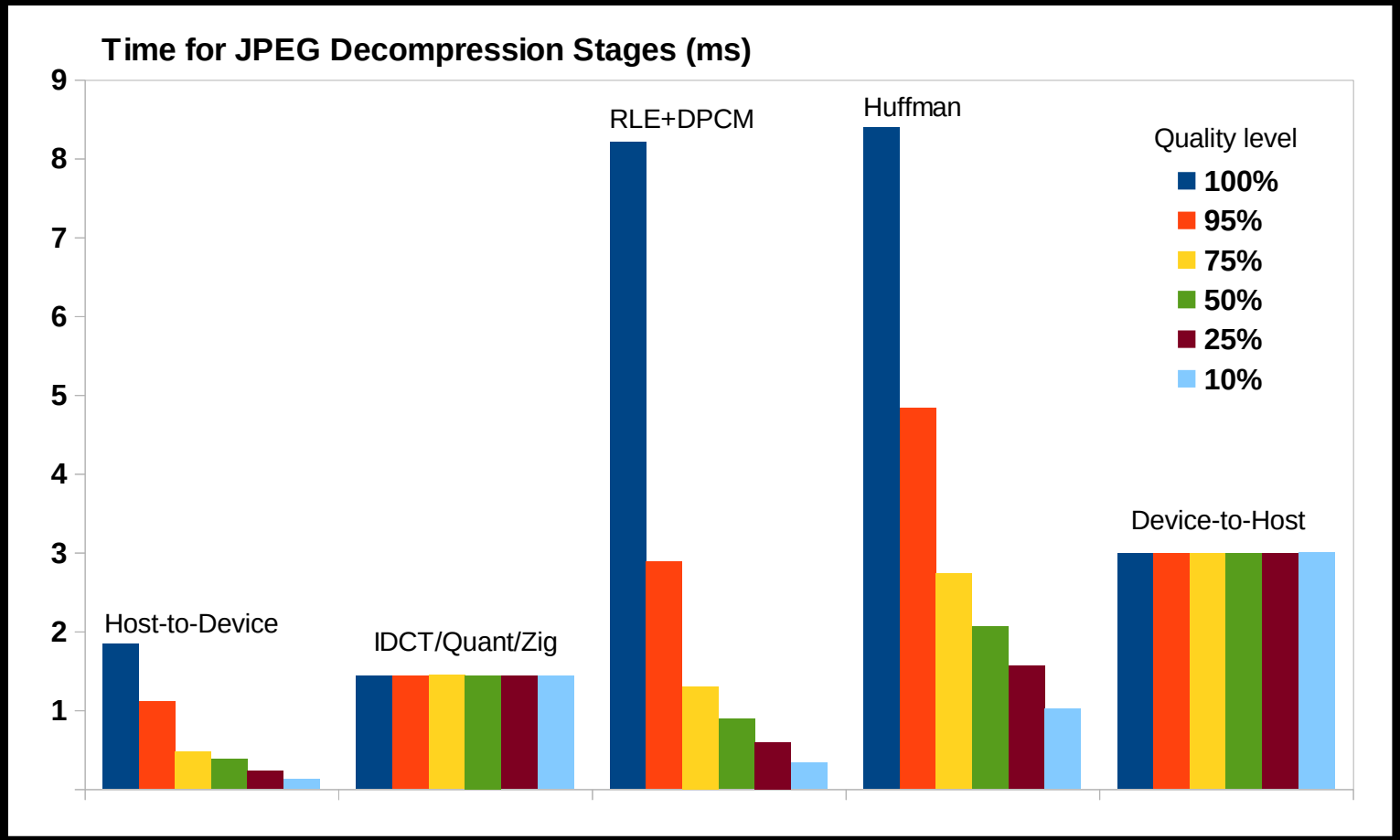
JPEG Decoding Rates for GPU & CPU



JPEG Decoding Time for GeForce GTX 580

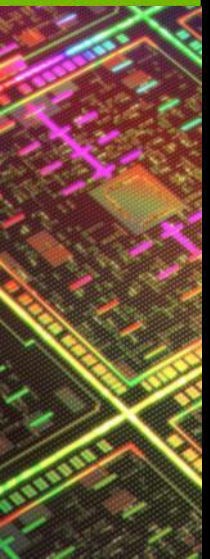


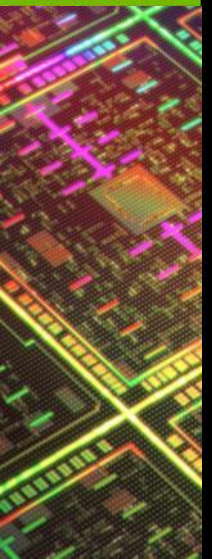
JPEG Decoding Time for GeForce GTX 680



Getting More Speed-up

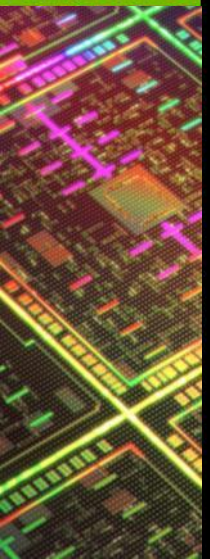
- GPUs with PCI-Express 3.0 interface
- Concurrent copy and execution
- Multi-GPU computing





Applications to 3D rendering

- Modern 3D applications are working with increasingly high-resolution data sets
- JPEG is a standard color map storage format
- Decoding JPEG on the CPU has major drawbacks
  - CPU-based decoding can be unacceptably slow even with partial GPU acceleration
  - Transferring raw decoded image or intermediate decoding results over PCI-Express is much more expensive
- JPEG decoding on the GPU is a perfect solution to both problems



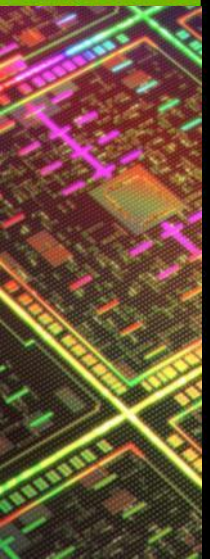
Applications to JPEG Imaging for Web

- Server-side image scaling to fit client devices.
- Thumbnail generation for big image databases.

Problem: how to cope with 100’s of millions images per day?

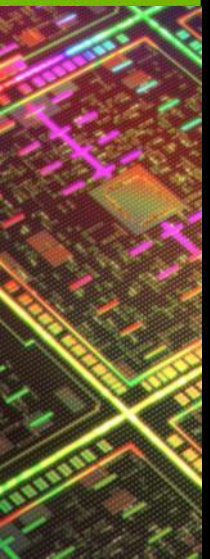
Method outline

- Get images from the database and load them to Host
- Image Decompression → Resize → Compression
- Store final images to the database or send them to users



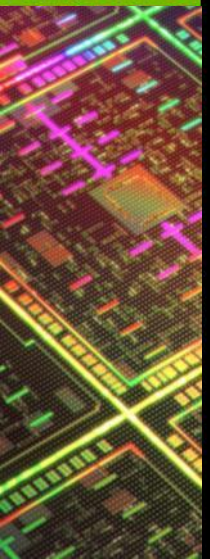
Conclusion

- Fast image coding on the GPU is reality
- Modern GPUs are capable of running many non-floating point algorithms efficiently



Future Work

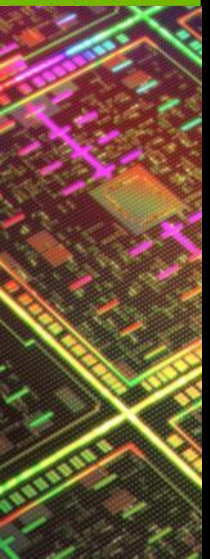
- SDK for FVJPEG codec for Windows / Linux
- Optimized JPEG, MJPEG, JPEG2000
- Multi-GPU computing
- Custom software design



Questions?

- Fyodor Serzhenko npoastek@mail.ru
- Victor Podlozhnyuk victor.podlozhnyuk@gmail.com

- More info at www.fastcompression.com

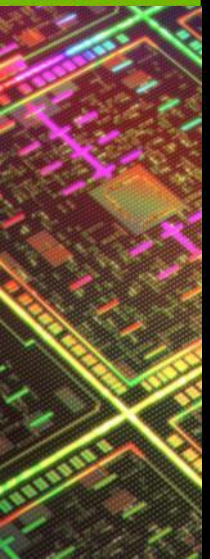


PCs & Laptop for testing

- ASUS P6T Deluxe V2 LGA1366, X58, Core i7 920, 2.67 GHz, DDR-III 6 GB, GPU GeForce GTX 580 or GeForce GT 240
- ASUS P8Z77-PRO, Z77, Core i7 3770, 3.4 GHz, DDR-III 8 GB, GPU GeForce GTX 680 (cc = 3.0, 1536 cores)
- OS Windows-7, 64-bit, CUDA 4.1, driver 296.10

Laptop

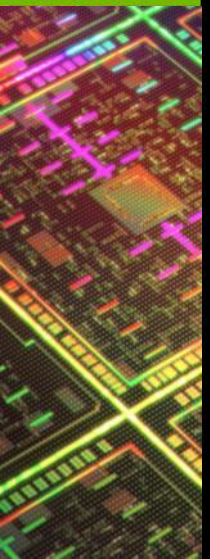
- ASUS N55S, Core i5 2430M, DDR III 6 GB
- GeForce GT 555M (cc = 2.1, 144 cores)
- OS Windows-7, 64-bit, CUDA 4.1, driver 296.10



Baseline JPEG parameters for test

- 8-bit grayscale images
- Compression quality from 10% to 100%
- Default static quantization and Huffman tables
- Test image: 7216 x 5408, 8-bit, CR = 12.8
- 8-thread encode/decode option for CPU

Conclusion: These parameters define the same calculation procedures for CPU & GPU.



Test image