

Development of a high-speed camera with integrated real-time video compression

Prof Dr Christoph Stamm of the School of Engineering at University of Applied Sciences and Arts Northwestern Switzerland (FHNW) and AOS Technologies AG, a leading manufacturer of high-speed imaging systems, have developed a prototype of a high-speed camera that operates at over 500 frames per second and has high definition resolution video compression. To accomplish this goal, the team of researchers has designed a complex coding sequence to compress the image frames, which has been programmed onto a specialised integrated circuit.

A contemporary problem in slow motion video production is that of obtaining high definition (HD) frames at sufficient speed on playback. Currently, the best high-speed cameras can produce HD video at over 500 frames per second. However, cameras are unable to transmit this amount of data over standard data interfaces at the same speed and so, the frames are stored on the camera's RAM memory. The maximum size of the video depends on the amount of storage on the camera, which can become expensive for longer video clips. An alternative way to solve this problem is the real-time compression of files before they are saved to memory.



Dr Stamm and his collaborative team have developed a new real-time compression/decompression (codec) algorithm to shrink the size of a video frame. This type of compression is common, but compression can often cause some loss of data that cannot be recovered upon decoding, resulting in a lower quality output image. The researchers were able to develop their codec for either lossless compression with small compression ratio or lossy compression with a good trade-off between image quality and compression ratio.

HIGH-SPEED PROGRAMMABLE CIRCUITS

The researchers designed their image compression algorithm to be programmed onto an integrated circuit called a Field Programmable Gate Array (FPGA), which can be programmed after manufacturing. FPGAs contain lots of small logic processing blocks and memory cells that can be lined up in parallel or in series such that they execute a complex function. Another type of integrated circuit called ASIC (Application Specific Integrated Circuits) were much more common in the past, however, FPGAs have become more common due to their flexibility and low initial cost thanks to important progress in terms of reduced power usage, increased speed, higher complexity and lower material cost. Mapping the algorithms onto the FPGA meant the team had to redesign the algorithm such that it operates in parallel, dramatically increasing the computation power compared to a software running on a microprocessor. The design had to be optimised such that it fits in the selected device, that it required a minimal amount of FPGA-internal memory and that it runs at the desired clock speed in order to reach the project targets of the codec.

When a code is mapped onto a programmable logic device such as an FPGA, then it is the actual hardware being configured, rather than the code being executed like software, and is, therefore, a more permanent type of

code. Consequently, the engineers first established and tested the codec by programming it onto a software reference model using the programming languages C++ and Open CL before implementing the algorithm onto the FPGA circuit using a different computing language called VHSIC Hardware Description Language (VHDL), a common computer language used for programmable hardware.

Nils Frey and Marcel Baier, graduate students under the supervision of Michael Pichler and Dino Zardet and supported by Dr Stamm, developed the design of the codec algorithms onto the series of blocks, thereby compressing the pixels of video frames in sequence. One of the features of the new codec algorithm is, that it does not require complete images fully available in memory but that they

The researchers designed their image compression algorithm to be programmed onto an integrated circuit called a Field Programmable Gate Array (FPGA), which can be programmed after manufacturing.

are streamed in from the sensor line by line, enabling high-quality images to be maintained and played back with great speed.

NOISE REMOVAL AND COLOUR TRANSFORMATION

When light first enters a camera, it is captured by a sensor with about two million pixels, which converts light waves into small junks of electrical charge (a number of electrons) that convey the information. Since the material and the manufacturing are not 100% constant over the whole array of pixels, small differences of the characteristics of the individual pixels form a so called Fixed Pattern Noise (FPN), which results in inconsistencies in the brightness of pixels at the same exposure. The removal of this type of noise is common in cameras as a first step before compression. For the camera prototype, the signal wiped free from noise, is sent to the colour transform (CT) block, the first step in compressing the image data. Sensors in video cameras detect light intensity

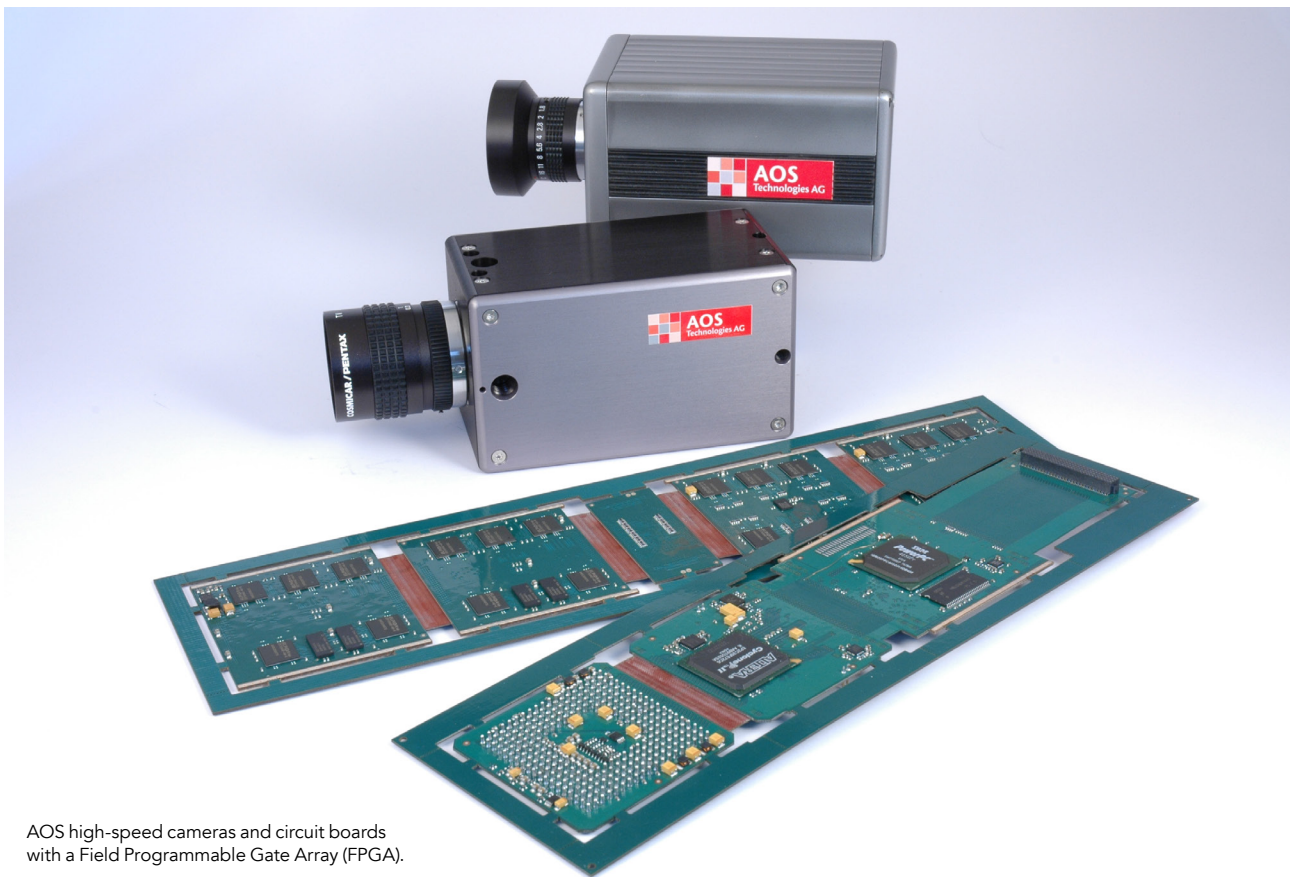
with little or no wavelength specificity and therefore cannot separate colour formation. Therefore, most use a so-called Bayer filter consisting of an array of many tiny colour filters placed over the image sensor to separate colour information. A Bayer filter pattern consists of 50% green, 25% red and 25% blue elements. The image sensor in the camera placed behind the Bayer filter captures and conveys information to the CT block in a format called Red-Green-Green-Blue (RGGB) format about the intensity of the light in these wavelengths. The RGGB signals are not efficient for storage and transmission, since they have a lot of redundancy, and so the CT block converts the pixels from RGGB into a different type of colour space called YUVD where Y stands for the luminance (brightness) of the image, U and V stand for the chrominance (colour information) in a two dimensional plane, and D stands for the difference between the two green pixels.

To reduce ('compress') the amount of data represented in the YUVD format, algorithms such as Discrete Wavelet

Transform, Quantization, Run Length Encoding and Huffman Encoding are applied on each of the four data channels of the YUVD colour-transformed pixels.

DISCRETE WAVELET TRANSFORM

Wavelets, a type of mathematical function currently being used for data compression, are used to separate between average and detail information from two-dimensional signals, such as images. The Discrete Wavelet Transform block, (DWT) unit is the most extensive part of the research group's codec. Here the codec uses lowpass and highpass filtering, which restrict signals to only those in the low (L) or high (H) frequency range to transform each colour-transformed pixel into a number, called a wavelet coefficient. The low- and high-pass filters are first applied horizontally to the input pixels and then applied again vertically. The resulting coefficients are then separated into four different sub-bands: one average sub-band (LL), and three detail sub-bands (LH, HL, HH). These four sub-bands together form a



AOS high-speed cameras and circuit boards with a Field Programmable Gate Array (FPGA).

level. The next wavelet transform step does the same decomposition on the LL sub-band on the last level. Using their software simulations, the researchers found that applying the DWT to the LL sub-band three times gave the best result.

QUANTIZATION SIGNAL PROCESSING FURTHER DECREASES BANDWIDTH

The next block in the codec algorithm uses a type of signal processing called Quantization, a common method in digital image compression. Data quantization is performed on the wavelet coefficients, reducing the amount of data needed to store similar approximations of the wavelet coefficient in order to reduce the total amount of data. A version of scalar quantization with uniform interval length combined with a threshold, called dead-zone, for values near zero is used. Thus, the quantization is achieved by compressing a range of values to a single value and setting all numbers near zero to the value zero. The width of the dead-zone is a good parameter to control the resulting compression ratio. Quantization is the only source of information loss in the codec. Omitting the quantization step results therefore in lossless compression.

FINAL COMPRESSION STEPS

The final block is the step where (quantized) wavelet coefficients become run length encoded (RLE) and Huffman encoded, which researcher Marcel Baier from the FHNW's School of Engineering found was the most challenging step in the FPGA design process. These two processes are executed simultaneously. The run length encoder is a form of compression that has been used in the transmission of television signals as far back as 1967. In run-length encoding, runs of identical data are stored as a single value with an indication of how often it occurred. For further compression, the values are Huffman encoded. Huffman encoded means that very often used values are described by very short codes and rarely used values by long codes held in a table on the FPGA.

THE RESULT AND FUTURE

As soon as all compression steps are completed, the data is converted into a stream of bytes and stored in memory just besides the FPGA in the camera. The format of this data stream is similar to the 'Progressive Graphics File' (PGF), a format that is comparable with the well known JPEG format but has a better image compression efficiency than JPEG. By using the image compression, the video camera is able to transmit high-quality images with much less amount of data in real-time, with very little or even no loss of data quality. Dr Christoph Stamm and collaborators have successfully transferred the software model onto the FPGA for a specific camera of AOS Technologies AG, but are able now to adapt this solution to any kind of sensor or camera.

Dr Stamm and his collaborative team have developed a new real-time compression/decompression (codec) algorithm to shrink the size of a video frame.



Behind the Bench

Prof Dr Christoph Stamm

E: christoph.stamm@fhnw.ch **T:** +41 56 202 78 32 **W:** <http://www.libpgf.org/>

W: <https://www.fhnw.ch/en/about-fhnw/schools/school-of-engineering>

W: <http://www.aostechnologies.com/high-speed-cameras-high-speed-video-cameras/>

Detail

Prof Dr Christoph Stamm
University of Applied Sciences Northwestern Switzerland
FHNW
School of Engineering
Bahnhofstrasse 6
CH-5210 Windisch
Switzerland

Bio

Professor Stamm studied computer science at HTL Brugg/Windisch and ETH Zurich. In 2001, he received his PhD in theoretical computer science at ETH Zurich. His main research interests include: parallel and real-time algorithms, image- and video-processing, data compression, image and video codecs and, computer vision.

Research Objectives

Professor Christoph Stamm's research aims to solve the complex problem of image and video data compression by designing algorithms that maintain high definition images and lead to high-speed, real-time data transmission. Combined with advances in hardware technology, this research has the potential to radically change the field of digital imaging, leading to increasingly specialised solutions in video technology.

Funding

Innosuisse – Swiss Innovation Agency: <https://www.innosuisse.ch/inno/en/home.html>

Collaborators

- University of Applied Sciences Northwestern Switzerland, FHNW
- FHNW School of Engineering
- People from IMVS (Codec development): <https://www.fhnw.ch/en/about-fhnw/schools/school-of-engineering/institutes/institute-of-mobile-and-distributed-systems/fast-data>
- People from IME (hardware implementation): <https://www.fhnw.ch/en/about-fhnw/schools/school-of-engineering/institutes/institute-of-microelectronics/digital-asic-fpga-design>
 - Michael Pichler
 - Dino Zardet
 - Nils Frey
 - Marcel Baier
- Industrial partner: AOS Technologies AG: <http://www.aostechnologies.com/high-speed-cameras-high-speed-video-cameras/>

References

- 1: N. Frey, "Real-time high-speed image compression on FPGA, master thesis," 2016.
- 2: A. T. AG, "About AOS Technologies AG," . [Online]. Available: <http://www.aostechnologies.com/company/>
- 3: Wikipedia, "Chroma subsampling," 2016. [Online]. Available: https://en.wikipedia.org/wiki/Chroma_subsampling
- 4: Wikipedia, "Discrete wavelet transform," 2016. [Online]. Available: https://en.wikipedia.org/wiki/Discrete_wavelet_transform
- 5: Wikipedia "Huffman Coding" 2018 [Online] Available: https://en.wikipedia.org/wiki/Huffman_coding
- 6: M. Baier "HighspeedCoder: Real-Time Highspeed Image Compression on FPGA, P7a, P7b" 2017.
- 7: C. Stamm, M. Pichler, D. Zardet, M. Baier "CTI project: High Speed Cameras with Integrated Video Compression"

Personal Response

You mention that the innovative part of the codec is the 'memory-efficient implementation of the multi-level wavelet transform, which only requires a few image lines of memory.' Can you explain what makes this part of the design so unique?

Typical implementations of multi-level wavelet transform build for each colour channel a pyramid of detail sub-bands with a LL sub-band on top of the pyramid. The whole pyramid occupies the same amount of memory as the original colour channel. The quantization and decoding steps are then applied after the pyramid has been fully built from the top of the pyramid down to the first detail sub-bands. Such an implementation is useful on a computer with a huge memory. However, on a FPGA with real-time requirements we try to avoid storing large amounts of data for deferred processing and try to compute all necessary codec processing steps as soon as possible to be ready for the next input data. It results in a highly parallel, and therefore very complex, implementation of all codec processing steps in a pipeline architecture (depicted in the following four step diagram):

