Available online at www.sciencedirect.com
ScienceDirect
Procedia Engineering 5 (2010) 208–211
Procedia Engineering
www.elsevier.com/locate/procedia

Proc. Eurosensors XXIV, September 5-8, 2010, Linz, Austria

An Efficient Hardware-Optimized Compression Algorithm for Wireless Capsule Endoscopy Image Transmission

Jef Thoné, Jeroen Verlinden, Robert Puers

ESAT-MICAS, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, Leuven 3001, Belgium

---

Abstract

A dedicated image compression algorithm was developed for wireless capsule endoscopy (WCE). The proposed approach reduces the amount image data without compromising the image quality. This allows an increase in image throughput to improve the real-time vision capabilities of a WCE system. The algorithm is based on a normalized Haar-wavelet transformation, aiming at a hardware-optimized implementation in a low-power FPGA or telemetry ASIC for wireless capsule endoscopes. Taking advantage of the limited color content of endoscopic images, this solution uses the YCbCr color space combined with efficient recoding of the wavelet transformation data. In this way adequate compression ratios at decent image qualities can be achieved in a hardware efficient thus low power implementation, outperforming existing hardware implemented algorithms.

Keywords: Capsule Endoscopy, Image Compression, low power FPGA

---

1. Introduction

Wireless capsule endoscopy (WCE) gains more and more popularity among Gastro-Intestinal (GI) specialists, because of increased chances of early detection of GI cancers, with more comfort for the patient [1, 2]. The limited image resolution and frame rates, and the lack of active locomotion still keep WCE from being a justified substitute for standard colono/gastroscopy [5]. As enabling technology, 3D inductive powering [3] allows the use of higher frame rates and higher resolution cameras.

A typical data rate calculation for a wireless endoscopic camera is shown in the example of Table 1. It is clear that when wirelessly transmitting image information through the body, the high information content related to high resolution - high frame rate images, requires a large bandwidth. Binary shift keying is still the most applied mod-

Table 1: Data rate calculation example for a typical wireless endoscopy application.

FR (framerate)	:	15 fps
R (resolution)	:	640×480 px
PD (pixel depth)	:	10 bit/px
CR (compression ratio)	:	25
DR (data rate)	:	$\frac{(FR \times R \times PD)}{CR} = 1.843$ Mbps

doi:10.1016/j.proeng.2010.09.084

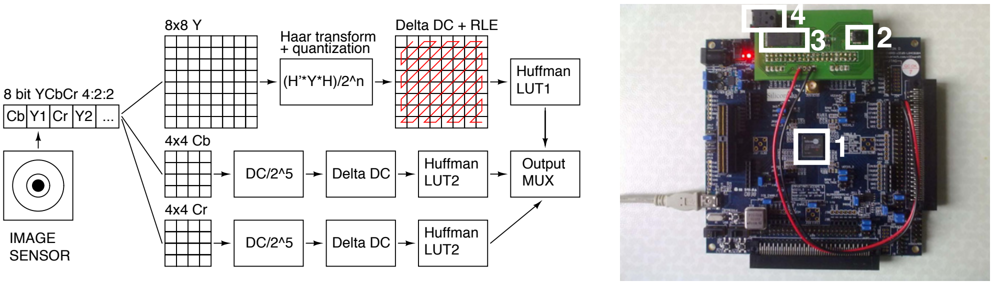

Figure 1: Functional block diagram of the proposed compression algorithm (left), and the prototyping board for wireless endoscopy imaging (right), with a Silicon Blue FPGA (1), TCM8230 VGA camera (2), SRAM (3) and Toslink optical connector (4).

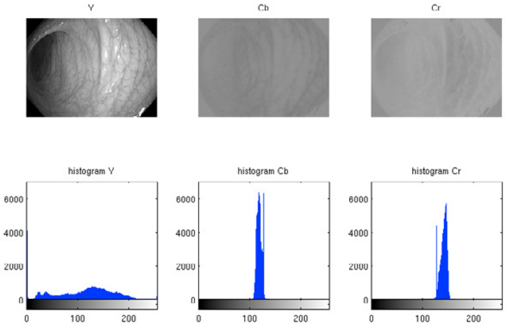

Figure 2: Histogram of the Y, Cb and Cr component of a typical endoscopic image. Note the limited information content for the Cb and Cr components.

ulation scheme in wireless endoscopy applications [4]. When opting for frequency shift keying (FSK) modulation, simple to implement and widely used because of its high robustness against noise, the required spectral bandwidth for transmitting raw image data can be estimated through Carson's rule [5]. Applied on the figures of Table 1, the transmitting antenna, receiving antenna and receiver require a bandwidth of at least 4 MHz. Higher image resolution, higher frame rate, error correction or reduced compression demand an even larger bandwidth.

It is clear from the previous observations that a trade-off has to be made between image quality, frame rate and compression ratio to be able to transmit within the available bandwidth. This paper describes a compression algorithm based on normalized Haar-wavelet transformation of the YCbCr color space, combined with efficient recoding to reduce the amount of data. This allows an increase in image throughput, improving the real-time vision qualities of a WCE system without compromising the image quality.

2. Compression Algorithm

The block diagram of the algorithm is presented in Figure 1. The image sensor is configured to output images in YCbCr 4:2:2 format, with Y the luma (or grayscale) component, Cb and Cr the blue-difference and red-difference chroma components. The 4:2:2 means that for each two Y components, one Cb and one Cr component is transmitted. Because of the limited color information in endoscopic images, the histograms of the Cb and Cr components are quite narrow compared to the Y component, as depicted in figure 2. The Cb and Cr component can be compressed more

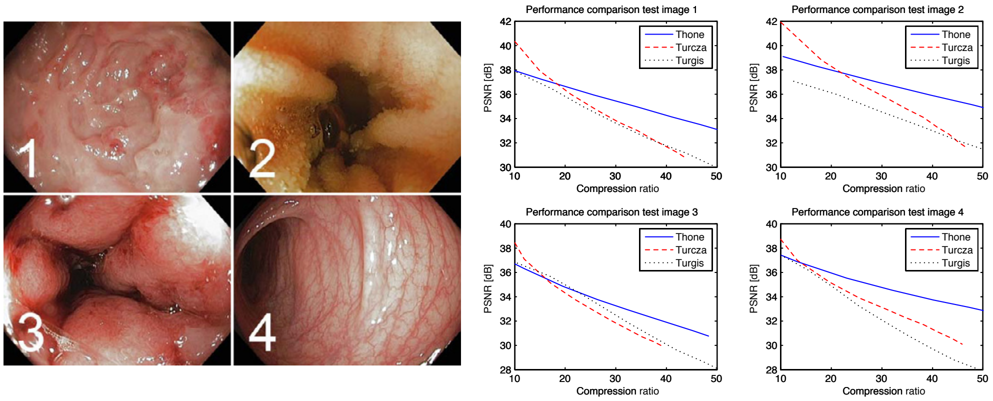

Figure 3: PSNR vs. compression ratio benchmarking of the proposed algorithm vs. the implementions of Turcza [8] and Turgis [7]. At CR > 25, the proposed algorithm proves to be superior. The set of test images is depicted on the left.

than the Y component, because of their smaller occurrence, and because of the limited sensitivity of the human eye to colour information. The Cb and Cr components are fragmented into 4x4 blocks, the sum of all the values (the DC value) in this block is calculated, and immediately quantized by removing the 5 least significant bits (LSB's). Next, the difference between subsequent DC-values (delta-coding) is used to select the right code from a fixed Huffman table for transmission. If the value has no matching symbol in the Huffman table, the code of escape symbol 'X' is used, followed by an 8 bit unsigned representation of the value + 128.

The Y component undergoes a more sophisticated compression. It is fragmented in 8x8 blocks, then a block-wise normalized Haar transform is performed. The Y and H matrix are multiplied, creating an intermediate matrix. The intermediate matrix and the transposed H matrix are multiplied, leading to the transformed coefficient matrix. The values are then quantized by removing n LSB's. N is user-defined to make the compression ratio adjustable.

The actual compression is achieved by standard zigzag run-length coding [6], delta coding of the DC coefficients followed by fixed Huffman table encoding. The run-length encoding follows a zigzag path through the coefficient matrix. Every sequence of zeros is replaced by a zero followed by the number of consecutive zeros. If the sequence runs until the end of the coefficient matrix, it is replaced by 2 zeros. Every DC-value at the top left of the coefficient matrix, is replaced by the difference with the DC-value of the previous block (delta coding). The resulting values are replaced by the matching Huffman code of another look-up table. If there is no matching Huffman code, the code of escape symbol 'X' is used followed by a 10 bit unsigned representation of the value + 512. The use of fixed Huffman tables has shown to have little influence on the compression ratio for the set of test images of Figure 3, and has the advantage of transmitting the images without their Huffman table. This slightly decreases the efficiency of the Huffman code, at the gain of an extremely simple and versatile Huffman implementation

Benchmarking. Matlab simulations on a set of 4 VGA test images show an increased performance of the proposed compression algorithm, compared to previously described implementations, especially for higher compression ratios, as depicted in Figure 3. The quality of a compressed image compared to the original is generally expressed in peak signal-to-noise ratio (PSNR). This is the ratio in dB between the maximal pixel value and the root-mean-square (RMS) error between the original and compressed image. Improvement in PSNR is possible at the lower compression ratios, when completely decomposing the Cb and Cr components. This has no added value however, as the required compression ratio is at least 20, where this algorithm outperforms other implementations [7, 8].

3. FPGA implementation

The prototyping environment is depicted on the right of Figure 1. Square 1 is a Silicon Blue 65L04 low power FPGA, 2 is a Toshiba TCM8230 VGA camera, 3 is a 2 Mbit SRAM and 4 is a Toslink optical connector. A PCB was developed to connect all the components to the FPGA evaluation board. The hardware implemented in the FPGA controls and read out the camera, compresses the images and sends the compressed data to a transmitter through the Toslink connection. The control of the camera is done by the I^2C bus. Initially, the 2 Mb RAM was used for raw image dumping, sending the image out at 2 Mbps [4], for verification of proper output data reading and data transmission. The implementation of the compression algorithm uses the FPGA's internal 80 kb of RAM. Although very small, it is dual ported and consist of 20 blocks of 4 kb which are all separately controllable.

The first VHDL process starts and sets the proper camera configuration through I^2C . Next, the camera is read out, also performing the first compression steps. The camera is configured as QQVGA (160 x 120 pixels), because of the limited available memory. The DC-values of the Cb and Cr blocks are calculated and stored in a separate 4 kb memory block. The intermediate matrix of the Haar-transform of the Y blocks is calculated line by line, and each line is stored in a separate RAM block, using 8 memory blocks in total. Doing so, the intermediate matrix is immediately transposed, allowing each column of the intermediate matrix to be accessed at once by simultaneously reading out all 8 memory blocks at the same address. When the first 8x8 intermediate matrix is finished, the second step of the Haar transformation is performed. The result is stored in another RAM block, and immediately quantized. Next, the full 8x8 coefficient matrix is zig-zag run-length encoded, while the DC values are delta-encoded and stored into an output RAM block. The output RAM is scanned by a separate transmitting process, cooperating closely with the Huffman encoding process. The scan process alternates between the Cb-Cr and Y output RAM blocks, retrieving its Huffman code and bit length, and finally transmitting it to the Toslink output at 2 Mbps. The transmitted data was received by a in-house developed USB interface board [9], for evaluation of the compression algorithm functionality.

4. Conclusion

A hardware efficient compression algorithm for endoscopic images is presented, outperforming existing hardware implementations for large compression ratios. The algorithm has been implemented in a CE prototype, and was functionally proven in a low power FPGA. This will allow a higher image throughput, improving the real-time vision capabilities of a WCE system, increasing acceptance by the medical community. Future improvements aim at miniaturizing the prototype to a pill-sized volume, including inductive powering and wireless transmission.

Acknowledgment

This work is supported by the European Community, within the 6th Framework Programme, through the Vector project (contract number 0339970). The authors would like to thank the project partners and the funding organization.

References

[1] C. M. Caffrey, O. Chevalerias, C. O. Mathuna, K. Twomey, Swallowable-capsule technology, Pervasive Computing 7 (1) (2008) 23–29.

[2] A. V. G. et al., Capsule endoscopy vs. colonoscopy for the detection of polyps and cancer, New England Journal of Medicine 361 (3) (2009) 264–270.

[3] R. Carta, G. Tortora, J. Thoné, B. Lenaerts, P. Valdastri, A. Menciassi, P. Dario, R. Puers, Wireless powering for a self-propelled and steerable endoscopic capsule for stomach inspection, Biosensors and Bioelectronics 25 (4) (2009) 845 – 851. doi:DOI: 10.1016/j.bios.2009.08.049.

[4] J. Thoné, S. Radiom, D. Turgis, R. Carta, G. Gielen, R. Puers, Design of a 2Mbps FSK near-field transmitter for wireless capsule endoscopy, Sensors and Actuators A: Physical 156 (1) (2009) 43 – 48.

[5] J. R. Carson, Notes on the theory of modulation, Proc. IRE 10 (1) (1922) 57–64.

[6] G. K. Wallace, The JPEG still picture compression standard, Commun. ACM 34 (4) (1991) 30–44. doi:http://doi.acm.org/10.1145/103085.103089.

[7] D. Turgis, R. Puers, Image compression in video radio transmission for capsule endoscopy, Sensors and Actuators A: Physical 123-124 (2005) 129 – 136.

[8] P. Turcza, T. Zielinski, M. Duplaga, Hardware implementation aspects of new low complexity image coding algorithm for wireless capsule endoscopy, in: Proceedings Computational Science ICCS 2008, 2008, pp. 476–485.

[9] D. Langsweirdt, Y. Berbers, J. Thoné, B. Puers, Design and implementation of a generic data-interface, in: Proc.IASTED Control and Applications, 2009, pp. 179–182.