

Research Article

Lossless and Low-Power Image Compressor for Wireless Capsule Endoscopy

Tareq Hasan Khan and Khan A. Wahid

Department of Electrical and Computer Engineering, University of Saskatchewan, 57 Campus Drive, Saskatoon, SK, Canada S7N 5A9

Correspondence should be addressed to Tareq Hasan Khan, tak488@mail.usask.ca

Received 10 February 2011; Accepted 12 April 2011

Academic Editor: Rached Tourki

Copyright © 2011 T. H. Khan and K. A. Wahid. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a lossless and low-complexity image compression algorithm for endoscopic images. The algorithm consists of a static prediction scheme and a combination of golomb-rice and unary encoding. It does not require any buffer memory and is suitable to work with any commercial low-power image sensors that output image pixels in raster-scan fashion. The proposed lossless algorithm has compression ratio of approximately 73% for endoscopic images. Compared to the existing lossless compression standard such as JPEG-LS, the proposed scheme has better compression ratio, lower computational complexity, and lesser memory requirement. The algorithm is implemented in a 0.18 μm CMOS technology and consumes 0.16 mm \times 0.16 mm silicon area and 18 μW of power when working at 2 frames per second.

1. Introduction

Wireless capsule endoscopy (WCE) [1–4] is a state-of-the-art technology to receive images of human intestine for medical diagnostics. In this technique, the patient ingests a specially designed electronic capsule which has imaging and wireless circuitry embedded inside (as shown in Figure 1). While the capsule travels through the gastrointestinal (GI) tract, it captures images and sends them wirelessly to an outside workstation (i.e., PC), where the images are reconstructed and displayed on a monitor for medical diagnostics. The development of wireless capsule endoscopy has changed video endoscopy of the little intestine into a much invasive and more complete examination. The increasing use of these resources and the comfort and ease with which some of these examinations can be performed makes it likely that wireless capsule video imaging will have a substantial impact on the management of small intestinal disease as well as other parts of the body. The capsule runs on button batteries that need to supply power for about 8–10 hours [1]. In this paper, our focus is on the image compressor in the capsule. Here, we propose an image compression algorithm by exploring the unique properties of endoscopic images. The scheme consists

of a simple and static prediction scheme and encoding the error both in golomb-rice [5, 6] and in unary coding. The algorithm is particularly suitable to work with any commercial low-power image sensors [7, 8] that output image pixels in raster scan fashion, eliminating the need of large buffer memory to store the complete image frame. The proposed algorithm has low computational complexity and it is simple to implement.

There have been some works reported on the image compressor of the capsule. In [9–11], Discrete-cosine-transform (DCT) based image compressors are proposed. In DCT-based image compressors, 4 \times 4 or 8 \times 8 pixel blocks need to be accessed from the image sensor. However, commercial CMOS image sensors [7, 8] send pixels in a row-by-row (i.e., raster-scan) fashion and do not provide buffer memory. So, to implement these DCT-based algorithms, buffer memory needs to be implemented inside the capsule to store an image frame. For instance, in order to store a 256 \times 256-size color image (i.e., 24 bits per pixel), a minimum size of 192 kB buffer memory is required. The memory takes large area and consumes sufficient amount of power which can be a noticeable overhead. Moreover, the computational cost associated in such transform-coding (i.e., multiplications,

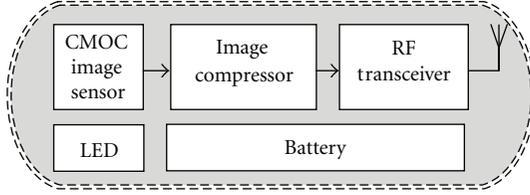


FIGURE 1: Block diagram of an endoscopic capsule.

additions, data scheduling, etc.) results in high area and power consumption. DCT-based compression algorithms are lossy. For proper medical diagnostics, lossless images are more desirable. Among lossless image compression standards, JPEG-LS [12, 13] can be a good choice for compression for endoscopic capsule application, because it can work with pixels coming in raster-scan fashion, does not need to buffer the whole image in memory. In [14, 15], design of an image compressor based on JPEG-LS algorithm is described. However, it needs memory to store at least one row of the image and approximately 1.9kB register arrays to store other key control parameters and contexts of JPEG-LS [16].

The rest of the paper is organized as follows. In Section 2, the design criterion of the image compressor is described. Section 3 describes several analysis of endoscopic images. In Section 4, the proposed compression algorithm and its performance is described. In Section 5, the complexity of the proposed algorithm is discussed. Section 6 discusses the hardware architecture of the compressor, and Section 7 concludes the paper.

2. Design Criterion

Considering the application need and the environment, we have set the following design criterion:

- (i) The operation of the capsule should be at low power consumption as the battery life is limited. Therefore, image-processing algorithms with high complexity (such as transform coding, wavelet decomposition, etc.) cannot be performed within the capsule. Hence, we focus on algorithms with very low complexity. This will give the opportunity to add additional features (e.g., robotic capability, speed control, [17–20], etc.) with the available capsules.
- (ii) The size of a typical capsule is limited to 28 mm × 11 mm [2]. Thus, the area is a critical issue that limits the usage of memory size. Memory consumes more silicon area and power. Here, we focus on algorithms that do not require much memory.
- (iii) The compressor should be able to work with commercially available CMOS image sensors which send data in raster-scan fashion.
- (iv) For an accurate diagnosis, the quality of image reconstruction is very important. So, we focus on lossless image compression algorithms.

- (v) Finally, the compression algorithm must also be able to reduce the data so that transmitter can send images at approximately 2 frames per second (fps) using its limited bandwidth.

3. Analysis of Endoscopic Images

In this section, we present our analysis on several endoscopic color images from [21]. One hundred test images have been taken from twenty different positions of the GI tract for the study; among them 20 images are shown in Figure 17.

3.1. RGB to YUV Conversion. First of all, we convert the test images from RGB color space to YUV using

$$\begin{aligned}
 Y &= \left(\frac{66 \times R + 129 \times G + 25 \times B + 128}{256} \right) + 16, \\
 U &= \left(\frac{-38 \times R - 74 \times G + 122 \times B + 128}{256} \right) + 128, \\
 V &= \left(\frac{112 \times R - 94 \times G - 18 \times B + 128}{256} \right) + 128.
 \end{aligned} \quad (1)$$

Figure 2 shows a 3D plot between component values (red, green and blue) and positions of an endoscopic image. From Figure 2, we can see that in RGB plane, the changes in pixel values are high, which means that there are more information contained in the three components. After the conversion to YUV, the changes in pixel values have been reduced as shown in Figure 3. In YUV, the intensity (luminance) is stored in Y component and U and V contains color information (chrominance). From Figure 3, the U and V component values do not vary much with positions and remain almost constant in the same plane. It is also noted that, this transformation does not theoretically degrade the quality of image; neither any information is lost. The small information content in U and V plane enables to use fewer bits to code them, without losing any information at all.

In addition, most commercially available image sensors are now equipped with a RGB-YUV converter. The proposed scheme aims to take the advantage of the on-chip converter, that is, there is no need to implement (1) inside the image compressor—that leads to save area and power consumption of the compressor.

3.2. Prediction. The JPEG compression standard supports a lossless mode where a simple lossless predictive coding (LPC) method is employed [5]. A predictor combines the values of up to three neighboring samples (A, B, and C) to form a prediction of the sample indicated by X in Figure 4.

Among the seven prediction modes as shown in Figure 5, the mode-1 prediction is the simplest to implement in hardware because pixels are sent in raster-scan fashion from image sensors and no buffer memory is required to implement mode-1 prediction scheme. On the other hand, mode-4 gives the most accurate prediction. Mode 4 has a geometrical interpretation as shown in Figure 6., which is, if we interpret each pixel as a point in a three-dimensional space, with the pixel's intensity as its height, then the value

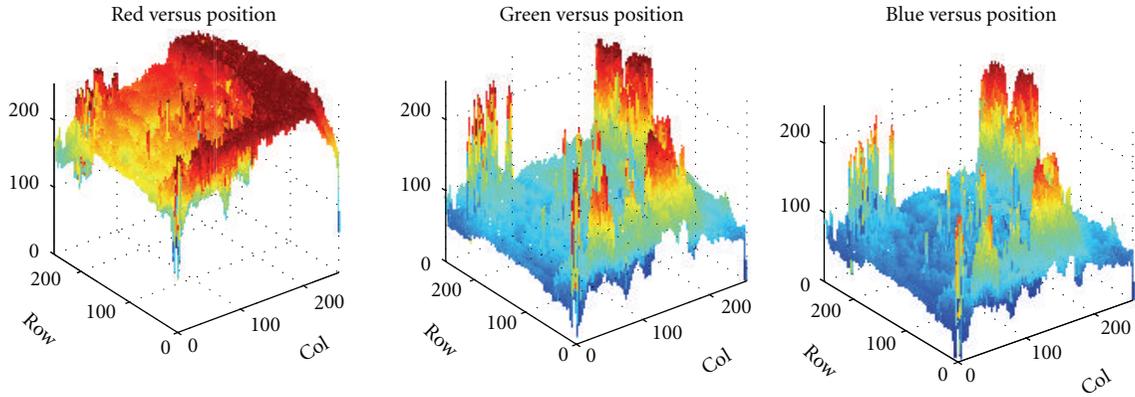


FIGURE 2: Red, green and blue components of an endoscopic image.

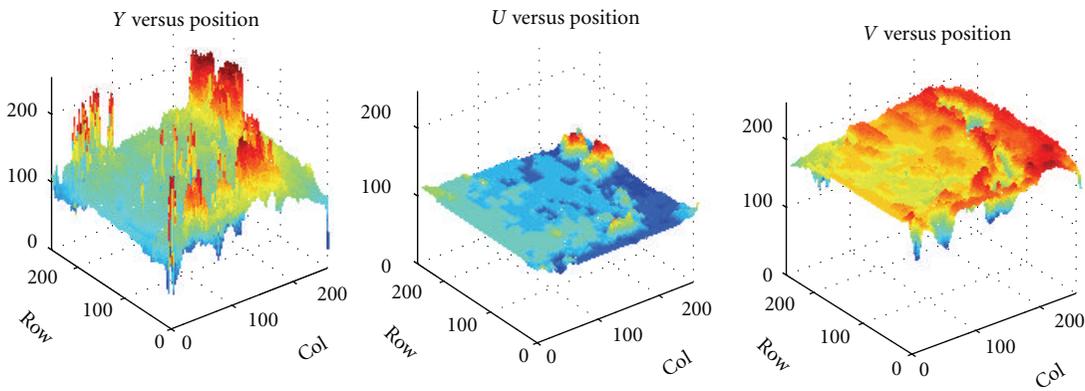


FIGURE 3: Luminance and chrominance components of an endoscopic image.

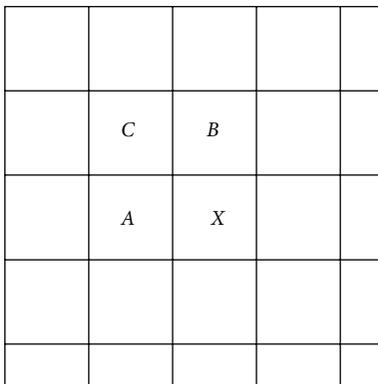


FIGURE 4: Prediction neighbourhood in lossless JPEG.

Mode	Prediction (X_p)	Mode	Prediction (X_p)
0	No prediction	4	$A + B - C$
1	A	5	$A + (B - C)/2$
2	B	6	$B + (A - C)/2$
3	C	7	$7(A + B) / 2$

FIGURE 5: Prediction modes.

$A + B - C$ places the prediction X_p on the same plane created by the pixels A , B , and C [5]. This produces the most accurate prediction when there is no sharp edge in the neighborhood. However, to implement mode-4 prediction scheme in hardware, the previous row of the current pixel needs to be buffered which increases the hardware cost.

For a comparative study, the compression ratio for each prediction mode is provided in Table 2.

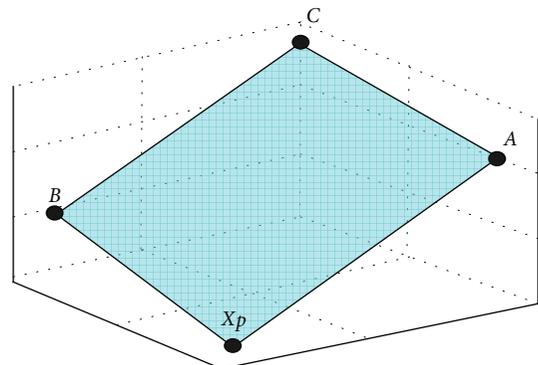


FIGURE 6: X_p falls on the same plane created by A , B , and C .

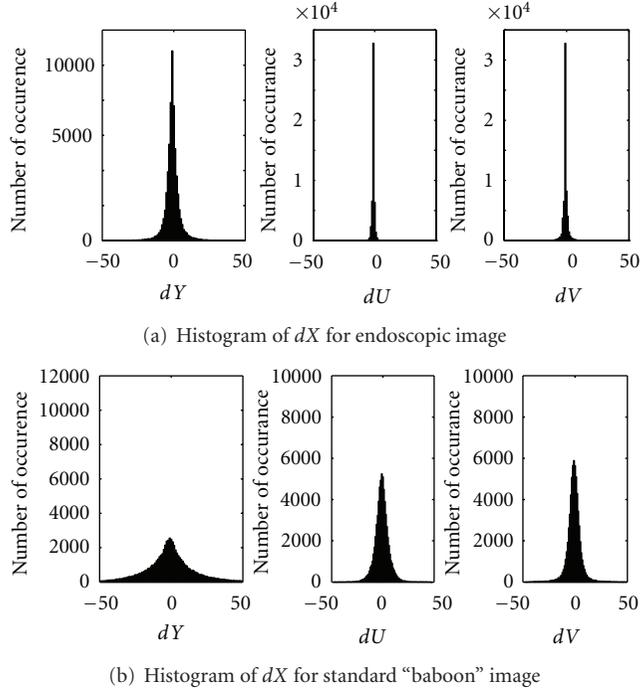


FIGURE 7: Histogram of dX for endoscopic and standard image.

The prediction X_p , is subtracted from the actual value of the sample X as shown in (2), and then the difference (dX) is entropy encoded.

$$dX = X - X_p \quad (2)$$

3.3. Difference Encoding. The next important step is to find a suitable variable-length encoding. For this purpose, we analyse the dX values of Y , U , and V components for model prediction. These changes (i.e., dY , dU , and dV) and their number of occurrence in an endoscopic image and an standard "baboon" image is shown in Figures 7(a) and 7(b) namely. Here, we see that for endoscopic images, the dU and also dV value occurs in a narrower range than standard images. It is due to the color homogeneity and the absence of bluish color objects in the GI tract of human intestine. The plots in Figure 7 shows two-sided *geometric distribution*.

3.3.1. Golomb-Rice Coding for dY Component. In the case of geometric distribution, the *golomb* code gives the optimum code length [6]. The *golomb-rice* code is a simpler version of the golomb code, which is easier to implement in hardware and have compression efficiency near to the golomb code [5]. Hence, we have chosen it to encode the difference values (dX). Since, dX can be either positive or negative, and golomb-rice code can work only with positive integers, we map the positive dX to *even integers* and negative dX to *odd integers* using (3). The mapped integers (m_dX) are then encoded using golomb-rice code.

$$m_dX = \begin{cases} 2dX, & \text{when } dX \geq 0 \\ 2|dX| - 1, & \text{when } dX < 0 \end{cases} \quad (3)$$

Our experiments show that the values of dY do not generally exceed the range from +127 to -128 due to the absence of extreme sharp changes between two consecutive pixels in endoscopic images. The dU and dV values vary in a narrower range. So, it can be safely assumed that, the mapped positive integers (m_dX) will range from 0 to 255, which can be expressed in binary using 8 bits. The optimized golomb-rice coding is as follows.

- (i) We denote the variable I using:

$$I = 2^8 = 256 \quad (4)$$

- (ii) The m_dX is divided by M , where M is a predefined integer, and a power of 2 as shown in (5):

$$M = 2^k,$$

$$q = \text{Integer} \left(\frac{m_dX}{M} \right), \quad (5)$$

$$r = m_dX \bmod M$$

- (iii) The quotient (q) is expressed in unary in $q+1$ number of bits. Then the remainder (r) is concatenated with the unary code, and r is expressed in binary in k number of bits. It is desirable to limit the size of the golomb-rice code as it becomes very long for larger values. This is done by using a parameter named *glimit*. If $q \geq (\text{glimit} - \log_2 I - 1)$, then the unary code of $\text{glimit} - \log_2 I - 1$ is generated. This acts as an escape code for the decoder and is followed by the binary representation of m_dX in $\log_2 I$ bits.

- (iv) The maximum length in golomb-rice code (*glimit*) is chosen as 32. The length of a golomb-rice code can be calculated using (6)

$$j = \text{glimit} - \log_2 I - 1,$$

$$gr_len = \begin{cases} q + 1 + k, & \text{when } q < j, \\ \text{glimit}, & \text{when } q \geq j. \end{cases} \quad (6)$$

The length of golomb-rice code for different values of k is shown in Figure 8. It has been noticed from Figure 7 that the most occurred value of dU and dV is zero and others are very close to zero. For dY , a wider range of value occurs. We choose $k = 2$ for encoding the mapped integers of dY .

3.3.2. Unary Coding for dU and dV Component. The number of occurrence of $dU = 0$ and $dV = 0$ is very high as shown in Figure 7. In golomb-rice code for $k = 2$, it needs 3 bits to represent "0". However, in *unary* coding, it only needs 1 bit to represent "0". Hence, to get a good compression, smaller length codes for zero and near-zero values need to be assigned. As dU and dV values are mostly zero (others are very near to zero), unary code can be used to get better compression. Unary code can be generated by setting $k = 0$ in the golomb-rice encoder. We use the maximum unary

TABLE 1: Coding scheme.

Variable	Coding scheme
m_dY	Golomb-rice ($k = 2$)
m_dU	Unary ($k = 0$)
m_dV	Unary ($k = 0$)

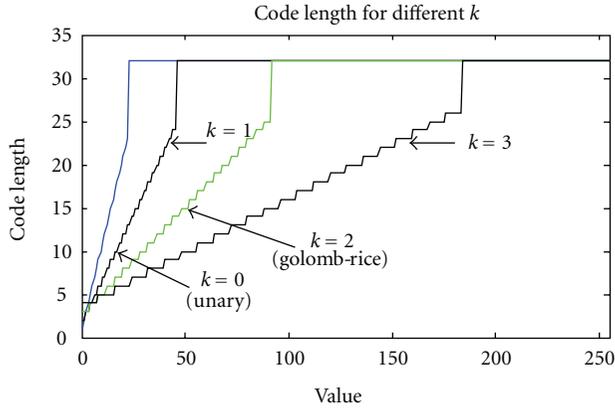


FIGURE 8: Length of golomb-rice and unary code.

code length ($ulimit$) as 32 similar as $glimit$ for golomb-rice code. It should be noted that, unary code gets very large for larger values as shown in Figure 8. dY has wider range of values than dU and dV as shown in Figure 7. So, unary coding of dY does not improve compression ratio. The coding scheme is summarized in Table 1.

3.4. Corner Clipping. WCE images are generally shown inside a round circular area as shown in Figure 9(a) due to the cylindrical shape of the GI tract. The black corners of the image do not have any importance in diagnostics. So, the capsule may discard these pixels during compression and thus higher compression ratio can be achieved. From hardware implantation point of view, it is easier to cut the four corners diagonally than cutting the image in a circular fashion.

For a square of length W , as shown in Figure 7(b), we calculated the maximum value of α using (7) where no pixels are discarded inside the circle of radius $W/2$.

$$\alpha = \frac{h}{\sin 45} = \sqrt{2} \left(d - \frac{w}{2} \right) = \frac{W}{2} (\sqrt{2} - 1) \times \sqrt{2} = 0.29 \times W \quad (7)$$

For a 256×256 image, the value of α is 75. Once, α is determined, the clipping algorithm can be implemented using a few combinational logic blocks as shown in Figure 10.

4. Proposed Compression Algorithm

Based on the above analysis, we have constructed a new image compression algorithm dedicated for capsule endoscopy application. The pixel values of the image are read (by the image compressor) in a raster-scan fashion. The overall algorithm is shown in block in Figure 11. The encoder

and the decoder using the proposed compression scheme have been implemented in PC software and verified.

We have applied the proposed compression algorithm with seven different prediction schemes on total one-hundred test endoscopic images of GI tract beginning from the larynx to the anus and the average compression ratio are shown in Table 2. The compression ratio (CR) is calculated using (8) and the overall peak-signal-to-noise-ratio (PSNR) is calculated using (9)

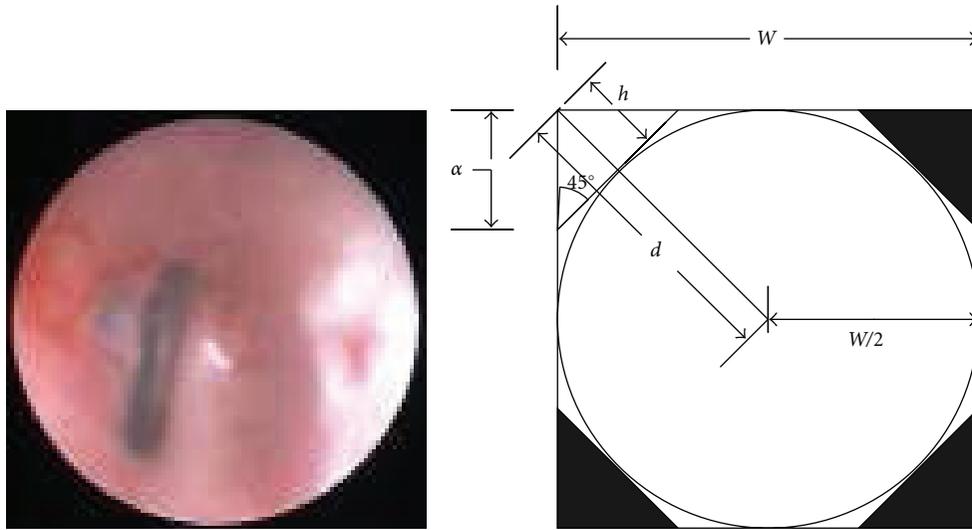
$$CR = \left(1 - \frac{\text{Total Bits After Compression}}{\text{Total Bits Before Compression}} \right) \times 100, \quad (8)$$

$$\begin{aligned} PSNR_{\text{overall}} &= 20 \times \log_{10} \\ &\times \frac{255}{\sqrt{(1/MxNx3) \sum_{c=1}^3 \sum_{n=1}^N \sum_{m=1}^M (x_{m,n,c} - x'_{m,n,c})^2}}, \end{aligned} \quad (9)$$

where M and N are the image width and height namely; x and x' are the original and reconstructed component values namely. C represents the three colour components red, green, and blue namely.

From Table 2, we see that the mode-4 prediction scheme gives the best compression ratio which is 72% without corner clip and 76.8% with corner clipping. The mode-1 prediction scheme is the simplest to implement in hardware as it does need buffer memory to store one row of image pixels. From Table 2, we see that the compression ratio for mode-1 is 67% without corner clipping and 73.2% with corner clipping. With an 800 kbps transceiver [22], compressed images can be transmitted at 1.9 and 2.2 frame per second using mode-1 and mode-4 prediction scheme namely. As the proposed compression is lossless, the reconstructed images are identical with the original image as shown in Figure 12. Hence, the overall average PSNR of the reconstructed images is infinity and the structural similarity (SSIM) index [23, 24] is 1.

In Table 2, comparison is also made with some existing works on endoscopic image compression. Here, we see that the proposed compression algorithm is the only lossless compression algorithm which produces infinite PSNR of the reconstructed images. In terms of compression ratio, the proposed algorithm outperforms [14, 15, 26, 27]. It should be noted that, the work in [9, 11, 25, 28] use DCT-based approach, which is computationally very expensive. Moreover, the reconstructed images may contain blocking artifacts due to inherent nature of DCT-based algorithms. Note that, our proposed scheme does not contain any blocking artifacts. Thus, considering the compromise of CR and PSNR, the proposed algorithm outperforms all other schemes by a good margin. The proposed lossless compression algorithm is applied on two standard images and the compression ratio is shown in Table 3. Comparing Tables 2 and 3, we see that the proposed compression algorithm works much better on WCE images than standard images. It is due to the fact that there are relatively large variations in U and V component in nonendoscopic images



(a) A typical endoscopy image

(b) Maximum length calculation

FIGURE 9: Clipping of image corners.

```

(1) Is_inside_visual_area:= False
(2) If (cY < alpha) {
(3)   If cX >= (alpha - cY) And cX < (W - (alpha - cY))
(4)     Is_inside_visual_area:= True }
(5) Else if cY >= (W - alpha) {
(6)   If cX >= ((cY - (W - alpha)) + 1) And cX < (W - ((cY - (W - alpha)) + 1))
(7)     Is_inside_visual_area:= True }
(8) Else
(9)   Is_inside_visual_area:= True
    
```

FIGURE 10: Pseudocode for corner clipping algorithm.

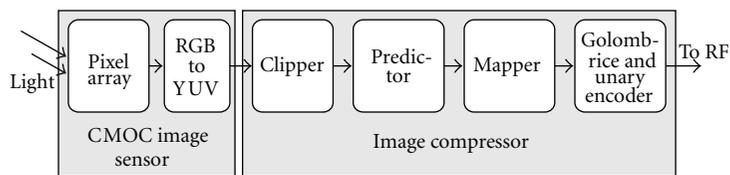


FIGURE 11: Block diagram of the proposed compression algorithm.



(a) Original

(b) Reconstructed

(c) Corner clipped

CR = 75.1% (mode-4),
CR = 72.7% (mode-1)

CR = 79.5% (mode-4),
CR = 77.9% (mode-1)

FIGURE 12: Original and reconstructed endoscopic images.

TABLE 2: Comparison of performance with other schemes.

Prediction mode	Average CR%		Overall PSNR (dB)
	Encoding (without clipping)	Encoding + Clipping	
Proposed	1	67.0	73.2
	2	66.3	72.6
	3	60.0	67.8
	4	72.0	76.8
	5	70.5	76.0
	6	70.1	75.7
	7	68.3	74.6
Chen et al. [14]		56.7	46.4
Xiang et al. [15]		72.7	46.8
Wahid et al. [9]		87.1	32.9
Turcza and Duplaga [10]		32.0	36.5
Lin et al. [11]		79.6	32.5
JPEG-FP [9, 25]		81.5	31.5
Hu et al. [26]		72.0 (Q level = 2)	39.6
Wu and Li [27]		50.0	31.0
Dung et al. [28]		82.0	36.2

TABLE 3: Compression ratio of standard images (without clipping).

Prediction mode->	1	2	3	4	5	6	7
Baboon	19.7	4.9	-10.9	25.3	28.8	24.4	25.2
Lena	47.0	55.7	36.4	61.1	57.4	60.6	55.2

TABLE 4: Comparison between proposed (mode-1) and standard JPEG-LS algorithm.

	Proposed algorithm	JPEG-LS
Color plane	YUV	RGB
Prediction	Static	Dynamic (based on edge detection)
Buffer memory requirement	0	memory to store one row of image pixel + 1.9 KB for context register
k -parameter	Fixed	Dynamically calculated based on context array
Coding	Golomb-rice and Unary	Golomb-rice
Run mode	No (as long runs are rare in WCE images)	Yes
Avg. CR for WCE Image	67.0% (without clip) 73.2% (with clip)	57.9% (without clip) —

than endoscopic images. As a result, dU and dV spans in wider range in nonendoscopic images as shown in Figure 7 and the use of unary code cannot produce good compression

ratio as unary code becomes very large for large values as shown in Figure 8.

5. Complexity Analysis

A comparison between the proposed lossless algorithm and the standard JPEG-LS is shown in Table 4. Here, we see that the proposed algorithm has lower computational complexity (such as static prediction and static k parameter) and lower memory requirement. We also see that the proposed compression algorithm outperforms the standard JPEG-LS algorithm [29] in compression ratio by 15% for endoscopic images. In Table 5, a comparison of the complexity of the proposed scheme with other works is shown. For an image of n pixels, the proposed algorithm has lowest computational complexity $O(n)$. For the mode-1 prediction scheme as shown in Figure 5, no buffer memory is required. However, for mode-4 prediction, buffer memory is required to store one row of the image pixels. On the other hand, the DCT-based algorithms [9–11, 25, 28] have complexity of $O(n \log n)$ and need memory buffer to store image frame.

Compared with the JPEG-LS scheme in [14], the presented algorithm implements simpler prediction scheme and works on YUV plane. Moreover, no buffer memory is required to store the context which is needed in standard JPEG-LS. The proposed algorithm does not implement the “Run mode” due to fact that endoscopic images do not contain long runs. The work in [26] uses a wavelet-based

TABLE 5: Comparison of complexity with other schemes.

	Type	Block size	Colour plane	Need of Buffer	Loss-less	Complexity
Chen et al. [14]	JPEG-LS (with run mode, $NEAR>0$)	—	RGB	Yes	No	$O(n)$
Xiang et al. [15]	JPEG-LS (with run mode, $NEAR>0$)	—	RGB	Yes	No	$O(n)$
Wahid et al. [9]	Transform coding (DCT)	4×4	RGB	Yes	No	$O(n \log n)$
Turcza and Duplaga [10]	Transform coding (DCT)	4×4	YCbCr	Yes	No	$O(n \log n)$
Lin et al. [11]	DCT and LZ77 encoding	8×8	RGB	Yes	No	$O(n \log n)$
JPEG-FP [9, 25]	Transform coding (DCT)	4×4	RGB	Yes	No	$O(n \log n)$
Hu et al. [26]	Wavelet transform coding	4×4	RGB	Yes	No	$O(n)$
Wu and Li [27]	Block-based compressed sensing	8×8	YUV	Yes	No	$O(n^3)$
Dung et al. [28]	H.264 based DCT	4×4	RGB	Yes	No	$O(n \log n)$
Proposed	Predictive coding	—	YUV	No	Yes	$O(n)$

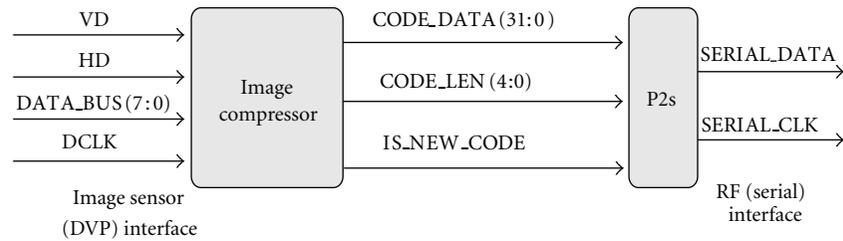


FIGURE 13: Block diagram of the compressor and P2S.

TABLE 6: Chip layout specification (mode-1 prediction).

Inputs/Outputs	13/2
Technology	0.18 μm CMOS
Die dimension ($W \times H$)	0.16 mm \times 0.16 mm
Core dimension ($W \times H$)	0.68 mm \times 0.68 mm
Number of cells	771
Latency	1 clock cycle
Max DCLK frequency	91 MHz
Core power consumption	287 μW @ fps = 2

coding; although the complexity of wavelet transform is $O(n)$, the actual implementation takes more power and area than the proposed predictive coding-based scheme. The work based on sensing theory [27] has the highest computational complexity, $O(n^3)$.

6. Hardware Architecture

The proposed lossless compression algorithm have been implemented in VHDL and simulated for functional verification. The compressor is designed to compress QVGA (320×240), 24 bits per pixel color images. Figure 13 shows the overall block diagram of the proposed compressor. The input lines are designed in a way that can be connected to any *digital-video-port* (DVP) based commercial image sensor [7, 8]. The compressor outputs 32-bit encoded bit vector and 5-bit code length. A parallel-to-serial (P2S) converter is needed to connect to commercial transceiver [22] which

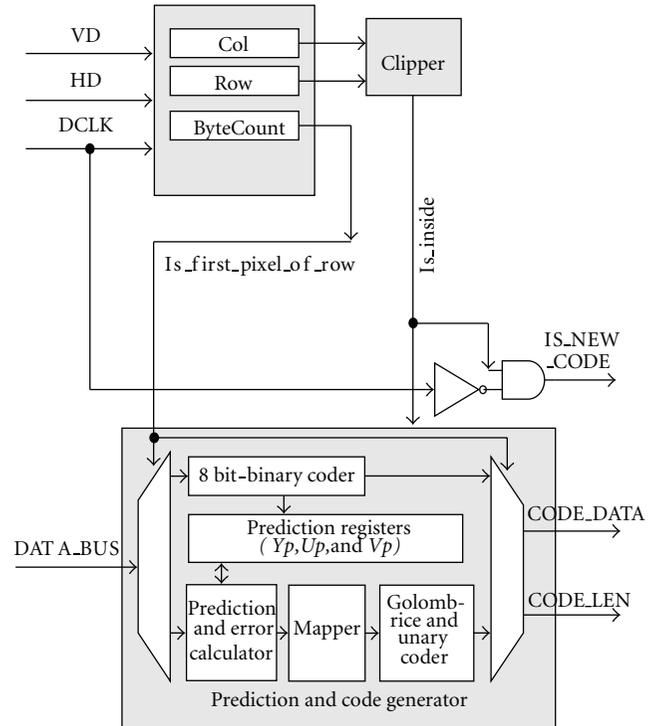


FIGURE 14: Block diagram of the image compressor.

accepts data serially using *serial peripheral interface* (SPI) (or any other serial) protocol.

Most commercial CMOS image sensors [7, 8] send image data bytes (both in RGB and YUV) in raster-scan fashion

TABLE 7: Comparison of Hardware cost with other schemes.

	Die size (mm × mm)	Core Size (mm ²)	Cell count	Gate count	Buffer memory	Latency (cc)	Power (mW)
[14]	3.4 × 3.3	11.2	—	19.5 K	2.19 KB	w^1	1.30 ²
[15]	3.0 × 4.2	—	—	90 K	103 KB	w^1	6.2
[9]	—	0.326	3,466	—	Yes	18	10.0
[11]	1.25 × 1.25	0.384	—	31 K	Yes	—	14.92
[28]	—	—	—	60 K	Yes	—	0.916
Our	0.68 × 0.68	0.0256	618	2 K	0	1	0.018

¹ w : width of input frame; for example., for a 256×256 image, the latency is 256; ²includes other digital components, such as microcontroller, I²C, and so forth.

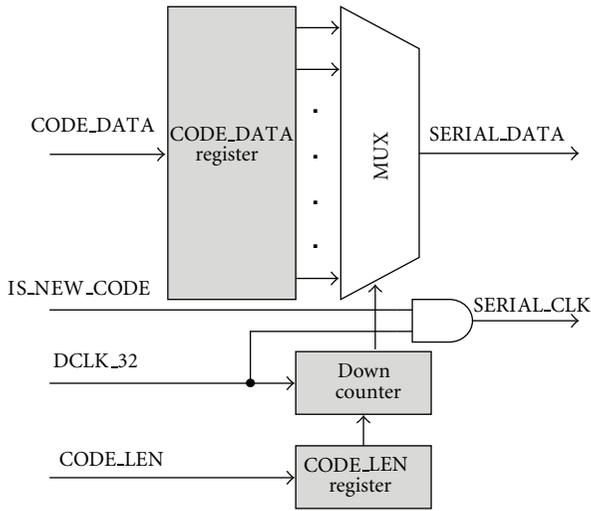


FIGURE 15: Block diagram of the parallel to serial converter (P2S).

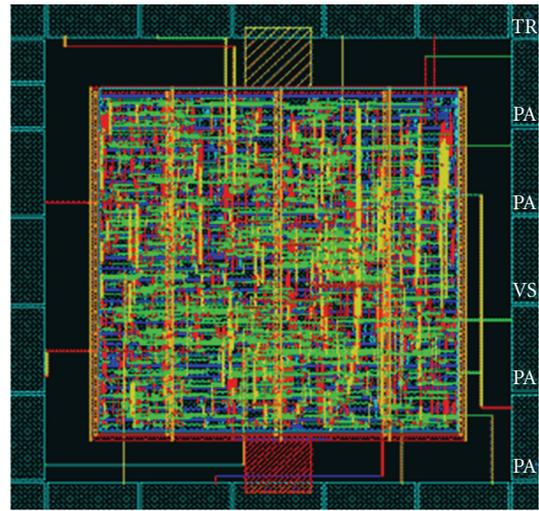


FIGURE 16: Chip layout of the lossless image compressor (with P2S).

TABLE 8: Chip layout specification (mode-4 prediction).

Inputs/Outputs	13/2
Technology	0.18 μ m CMOS
Die dimension ($W \times H$)	1.6 mm × 1.6 mm
Core dimension ($W \times H$)	1 mm × 1 mm
Number of cells	24 K ¹
Latency	1 clock cycle
Max DCLK frequency	42 MHz
Core power consumption	1.78 mW ² @ fps = 2

^{1,2}Including the buffer memory to store one row of images for Y, U, and V components.

using a common DVP interface [30]. The input lines of the proposed compressor implements the DVP interface so that commercial CMOS image sensors can be directly interfaced with the proposed compressor. In a DVP interface, the VD (or VSYNC) and HD (or HSYNC) pins indicate the end of frame and end of row respectively. The image compressor samples the YUV pixel bytes from DATA_BUS(7:0) bus on each positive edge of DCLK and then generates the compressed variable length codes on the CODE_DATA(31:0) bus and its end bit index on CODE_LEN (4:0) bus within

a single clock cycle (cc) of DCLK. Thus, the latency of the proposed design is 1 cc. Whenever a new code appears on the data bus, the IS_NEW_CODE signal goes from low to high. Compressed output bitstream is available for sampling at the SERIAL_DATA pin at each positive edge of SERIAL_CLK.

6.1. Image Compressor. The hardware architecture of the image compressor is shown in Figure 14. The position registers such as the Col, Row, and ByteCount stores the column, row, and byte position in the row of the currently sampled pixel namely. From the Col and Row, the Clipper module decides whether clipping is necessary. If the pixel is inside the defined visual region, then the mode-1 prediction is made and subtracted from the current pixel value. Then the error is mapped to positive integer and a variable length golomb-rice code for Y component and variable length unary code for U and V component is generated along with the code length.

6.2. Parallel to Serial Converter (P2S). The architecture of the P2S block is shown in Figure 15.

It samples the CODE_DATA and CODE_LEN buses at the low to high transition of the IS_NEW_CODE and then

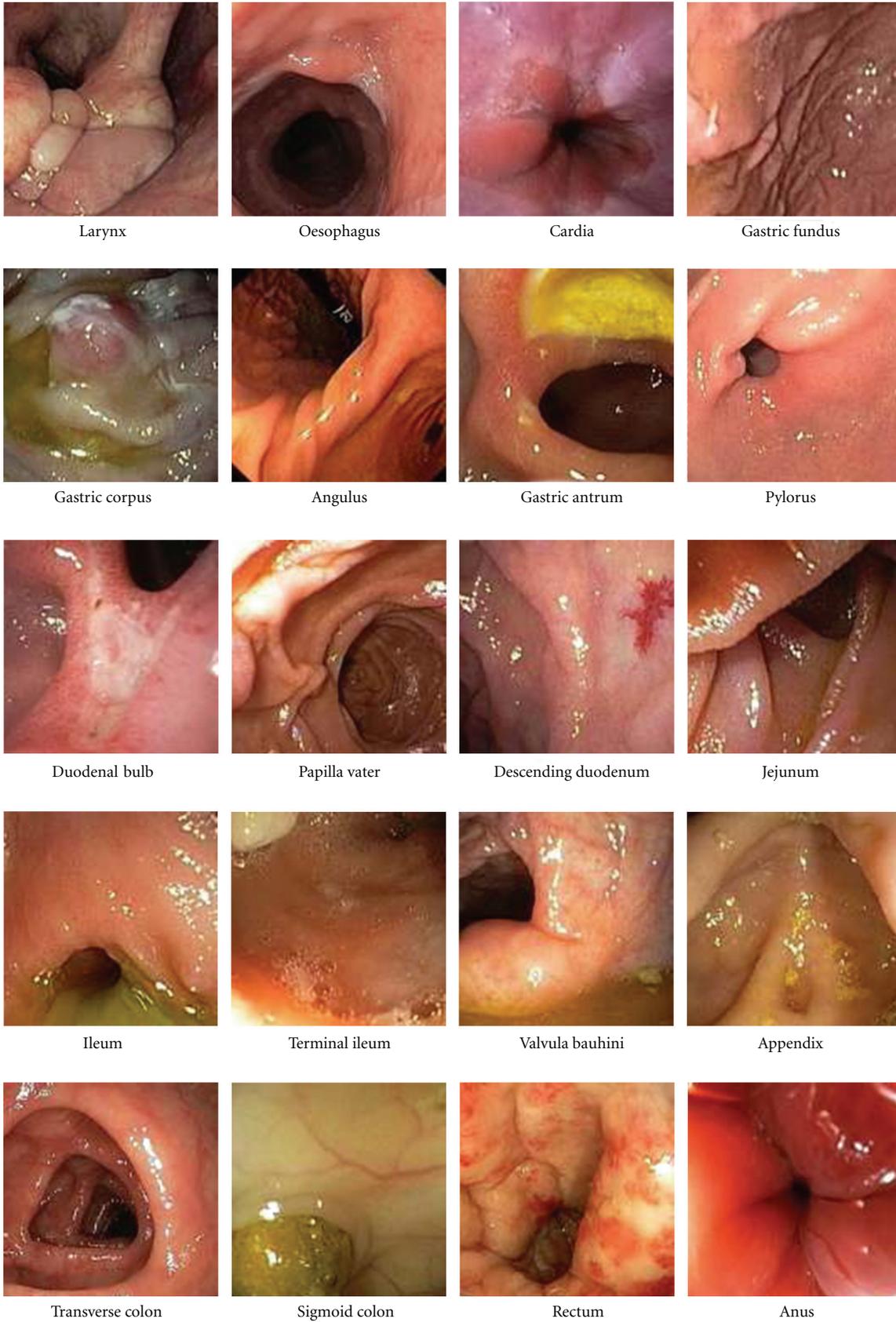


FIGURE 17: 20 (among 100) test endoscopic color images.

sends the variable length code bitstream serially (starting MSB first) using the *SERIAL_DATA* and the *SERIAL_CLK* pins. The *DCLK_32* is the input clock signal for this module that has at least 32 times higher frequency than the *DCLK* frequency, so that in one clock cycle of *DCLK*, the maximum length code (which is $glimit = ulimit = 32$) can be safely transmitted serially. At each positive edge of *SERIAL_CLK*, the RF transceiver can sample the bits from the *SERIAL_DATA* pin.

The overall design including the P2S module is synthesized using $0.18\ \mu\text{m}$ CMOS technology using standard Artisan library cells. Figure 16 shows the chip layout of the compressor. The chip specification is presented in Table 6. For a better assessment of the proposed compression scheme, the P2S converter (which merely facilitates the interfacing with RF transceiver) should not be considered as part of the compressor. Then the image compressor core itself (without the P2S module) consumes 618 cells and $18\ \mu\text{W}$ of power at 2 fps.

In Table 7, we compare the cost of implementation with other existing schemes. Here, we see that the proposed compressor occupies the least core area and hardware cost (i.e., gate count). It does not require any memory buffer for temporary storage of image frame or blocks. The power consumption of the compressor is also the lowest (less than 1mW) comparing with all other works. The latency for [11, 28] were not reported. However, these schemes use 2-D 8×8 DCT coding that is similar to the DCT implementations presented in [31, 32]. As a result, the latency for [11, 28] is estimated to be in the range from 92 to 144 (from [31, 32] resp.). Compared to all these existing designs, the proposed scheme has a latency of 1 cc which is the lowest. Thus, the scheme meets all the design criterion set in Section 2 and presents itself as a strong candidate for energy-efficient implantable lossless image compressor for wireless endoscopic applications.

We have also implemented the image compressor with mode-4 prediction which gives the best compression ratio. However, one row of image pixels for the 3 color components needs to be buffered for mode-4 prediction. Table 8 shows the chip specification with mode-4 prediction. By comparing Table 6 and 8, we see that the hardware cost is much higher in mode-4 prediction. It should also be noted that by using *memory compiler* synthesis tool, the hardware cost can be reduced.

7. Conclusion

In this paper, a lossless image compression algorithm for endoscopic images has been presented. The algorithm consists of a static prediction scheme and encoding the error both in golomb-rice and in unary code. The algorithm is suitable to work with any commercial low-power image sensors that outputs image pixels in a raster scan fashion, eliminating the need of buffer memory. The proposed lossless compression algorithm is implemented in a $0.18\ \mu\text{m}$ CMOS technology and consumes $0.16\ \text{mm} \times 0.16\ \text{mm}$ silicon area, and $18\ \mu\text{W}$ of power when working at 2 frames per second.

Acknowledgments

The authors would like to acknowledge the Natural Science and Engineering Research Council of Canada (NSERC) for its support to this research work. The authors are also indebted to the Canadian Microelectronics Corporation (CMC) for providing the hardware and software infrastructure used in the development of this design.

References

- [1] A. Moglia, A. Menciassi, and P. Dario, "Recent patents on wireless capsule endoscopy," *Recent Patents on Biomedical Engineering*, vol. 1, no. 1, pp. 24–33, 2008.
- [2] D. O. Faigel and D. R. Cave, *Capsule Endoscopy*, Saunders Elsevier, Amsterdam, The Netherlands, 2008.
- [3] P. Swain, "The future of wireless capsule endoscopy," *World Journal of Gastroenterology*, vol. 14, no. 26, pp. 4142–4145, 2008.
- [4] C. McCaffrey, O. Chevalerias, C. O'Mathuna, and K. Twomey, "Swallowable-capsule technology," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 23–29, 2008.
- [5] D. Salomon, *Data Compression: The Complete Reference*, Springer, 3rd edition, 2004.
- [6] S. Golomb, "Run-length encodings," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, 1966.
- [7] "OmniVisoin OVM7690 cameracube," 2010, <http://www.ovt.com/>.
- [8] "Aptina MT9V131 image sensor," 2010, <http://www.aptna.com/>.
- [9] K. Wahid, S. B. Ko, and D. Teng, "Efficient hardware implementation of an image compressor for wireless capsule endoscopy applications," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 2761–2765, June 2008.
- [10] P. Turcza and M. Duplaga, "Low-power image compression for wireless capsule endoscopy," in *Proceedings of the IEEE International Workshop on Imaging Systems and Techniques*, pp. 1–4, May 2007.
- [11] M. Lin, L. R. Dung, and P. K. Weng, "An ultra-low-power image compressor for capsule endoscope," *BioMedical Engineering Online*, vol. 5, no. 14, pp. 1–8, 2006.
- [12] ITU-T Recommendation T.87, "Lossless and near-lossless compression of continuous-tone still images—Baseline," 1998.
- [13] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.
- [14] X. Chen, X. Zhang, L. Zhang et al., "A wireless capsule endoscope system with low power controlling and processing ASIC," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 1, pp. 11–22, 2009.
- [15] X. Xiang, L. Guolin, C. Xinkai, L. Xiaowen, and Z. Wang, "A low-power digital IC design inside the wireless endoscopic capsule," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 11, pp. 2390–2400, 2006.
- [16] M. Papadonikolakis, V. Pantazis, and A. P. Kakarountas, "Efficient high-performance ASIC implementation of JPEG-LS encoder," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, pp. 1–6, April 2007.
- [17] N. Bourbakis, G. Giakos, and A. Karargyris, "Design of new-generation robotic capsules for therapeutic and diagnostic

- endoscopy,” in *Proceedings of the IEEE International Conference on Imaging Systems and Techniques*, pp. 1–6, 2010.
- [18] M. Quirini, A. Menciassi, C. Stefanini, S. Gorini, G. Pernorio, and P. Dario, “Development of a legged capsule for the gastrointestinal tract: an experimental set-up,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 161–167, July 2005.
- [19] S. Hosseini and M. B. Khamesee, “Design and control of a magnetically driven capsule-robot for endoscopy and drug delivery,” in *Proceedings of the IEEE Toronto International Conference Science and Technology for Humanity*, pp. 697–702, September 2009.
- [20] O. Alonso, L. Freixas, and A. Dieguez, “Advancing towards smart endoscopy with specific electronics to enable locomotion and focusing capabilities in a wireless endoscopic capsule robot,” in *Proceedings of the IEEE Biomedical Circuits and Systems Conference*, pp. 213–216, November 2009.
- [21] “Gastrolab,” 2010, <http://www.gastrolab.net/>.
- [22] Zarlink Semiconductor, “ZL70101 MICS transceiver,” 2010, <http://www.zarlink.com/>.
- [23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [24] Z. Wang and A. C. Bovik, “Mean squared error: love it or leave it?” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [25] C. Cheng, Z. Liu, C. Hu, and M. Meng, “A novel wireless capsule endoscope with JPEG compression engine,” in *Proceedings of the IEEE International Conference on Automation and Logistics*, pp. 553–558, August 2010.
- [26] C. Hu, M. Meng, L. Liu, Y. Pan, and Z. Liu, “Image representation and compression for capsule endoscope robot,” in *Proceedings of the IEEE International Conference on Information and Automation*, pp. 506–511, June 2009.
- [27] J. Wu and Y. E. Li, “Low-complexity video compression for capsule endoscope based on compressed sensing theory,” in *Proceedings of the IEEE Engineering in Medicine and Biology Society*, pp. 3727–3730, USA, September 2009.
- [28] L. R. Dung, Y. Y. Wu, H. C. Lai, and P. K. Weng, “A modified H.264 Intra-frame video encoder for capsule endoscope,” in *Proceedings of the IEEE Biomedical Circuits and Systems Conference (BIOCAS '08)*, pp. 61–64, November 2008.
- [29] “JPEG-LS public domain code,” 2010, <http://www.stat.columbia.edu/~jakulin/jpeg-ls/mirror.htm>.
- [30] T. Khan and K. Wahid, “Design of a DVP compatible bridge to randomly access pixels of high speed image sensors,” in *Proceedings of the IEEE International Conference on Consumer Electronics*, pp. 905–906, 2011.
- [31] T. Silva, C. Diniz, J. Vortmann, L. Agostini, A. Susin, and S. Bampi, “A pipelined 8×8 2-D forward DCT hardware architecture for H.264/AVC high profile encoder,” in *Proceedings of the 2nd Pacific Rim Conference on Advances in Image and Video Technology*, pp. 5–15, Santiago, Chile, 2007.
- [32] L. Pillai, “Video compression using DCT,” 2002, *Xilinx Application Note: Virtex-II Series*.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

