Ročník 2013 electroscope Číslo V

Fast and Power Reduced RNS-Based Image Filtering in Spatial Domain

D. Younes, P. Steffan
Department of Microelectronics, Faculty of Electrical Engineering and Communication, Brno University of Technology, Technická 3058/10, Královo Pole, 61600, Brno
E-mail : xyoune00@stud.feec.vutbr.cz, steffan@feec.vutbr.cz

Abstract:
This paper presents a fast and power reduced design for digital image filtering. Contrary to the typical designs for image processing, this one is based on a non-weighted number system, residue number system (RNS). The proposed structure applies a number of spatial filters on grayscale images. All the processing is done using the RNS, which results in faster and power efficient performance than those based on the standard binary number system.
This paper also presents detailed comparisons with other already published papers that use the RNS for digital image processing purposes. However, these papers used improper moduli sets, hence their filtering processes are inaccurate. Contrary to those designs, the proposed one performs the filtering process accurately. The filtered images are identical to those based on the binary number system.

INTRODUCTION

The carry-free and parallelism features of the residue number system (RNS) can be of a great benefit if utilized in suitable applications. The RNS divides the computations into a number of parallel channels according to the number of moduli. These channels are totally independent and perform high-speed computations on smaller residues [1], [2]. Moreover, the RNS is used in low-power designs. According to [3], the power dissipation is reduced by taking the advantage of the speed-up due to the parallelism of the RNS structure. Thus, the RNS has become a tough candidate for high-performance, low power, fault tolerant and secure digital signal processing (DSP) applications. This system has been intensively used in applications where addition, subtraction and multiplication are dominant. One of these applications is digital image processing.
Many researches were dedicated for exploiting the RNS features for enhancing digital image processing applications [4]–[8]. Each of them proposed its own structure. One of the first papers that suggested using the RNS in image processing is [4]. However, their paper was concentrated on the security concept rather than benefiting from the parallelism feature of the RNS. They used look-up tables (LUTs) for encoding and decoding, but their approach encrypts only a part of the image. Thus, [5] suggested an enhanced structure that encrypts the whole image. According to [5], the proposed scheme, which only consists of standard RNS blocks, offers high-speed and low-power implementation for secure image processing. An RNS based application for filtering digital images was presented in [6]. The filtering is done in both spatial and frequency domains. Since pixel values have the range [0,255], the authors suggested using the moduli set {5,7,8} as it provides a dynamic range [0,279], which they considered to be enough for image filtering applications. However, during our research, we have found out that this is not true, and presented an example that clarifies this confusion. In [7] and [8], similar structures for edge detection and spatial filtering were introduced. Again, the used moduli set here was {5,7,8}. All of these papers concentrated on timing performance of the proposed structures. They also mentioned that these applications have low power consumption. However, none of them reported the definite power consumption and compared it with the one based on the binary number system.
In this paper, we present a structure for implementing a digital image processing application using the RNS. This structure provides high-speed and power-reduced performance compared to the one based on the standard binary number system. The proposed structure has been implemented on Virtex-4 FPGA device. Both timing-performance and power consumption are reported and compared to those based on the standard binary number system.

RNS OVERVIEW

The RNS is defined by a set of positive pairwise relatively prime numbers $\{m_1, m_2,\ldots, m_n\}$ called moduli. In this system, each weighted number X is uniquely represented by an ordered set of residues $(x_1, x_2,\ldots, x_n)$. Each residue x_i is represented by,

$$x_i = X \bmod m_i = \langle X \rangle_{m_i} \quad ; \quad 0 \le x_i < m_i \tag{1}$$

The dynamic range of this system is defined as $M = m_1 \times m_2 \times \ldots \times m_n$. The range of representable integers is,

$$X \in [0, M-1] \tag{2}$$

For signed RNS, the range of representable integers is partitioned into two equal intervals,

$$\begin{aligned} &0 \le X < \lfloor M/2 \rfloor && \textit{for positive numbers} \\ &\lfloor M/2 \rfloor \le X < M && \textit{for negative numbers} \end{aligned} \qquad (3)$$

This means, if an output value belongs to the second interval, then its actual value is negative and is calculated by,

$$Actual(X) = X - M \qquad (4)$$

In this system, arithmetic operations (addition, subtraction and multiplication) are performed totally in parallel on those very independent residues.

$$X \circ Y = \left\{ \begin{matrix} \langle x_1 \circ y_1 \rangle, \langle x_2 \circ y_2 \rangle, \\ , \ldots, \langle x_n \circ y_n \rangle \end{matrix} \right\}; \ \circ \equiv \sim (+,-,\times) \qquad (5)$$

However, other arithmetic operations as divition, comparison, overflow and sign detection are problematic and complex.

A residue number can be converted back into its binary equivalent, by using one of the residue-to-binary conversion algorithms, such as, the Chinese Remainder Theorem (CRT), the Mixed-Radix Conversion (MRC), the new CRT-I, the new CRT-II, etc. [1], [2].

THE PROPOSED IMAGE FILTERING DESIGN

The proposed design applies a number of filters on a grayscale image. All the processing required for the filtering process is not performed using the standard binary number system but using the RNS. Since applying spatial filters demands performing many additions and multiplications, using the RNS can be of a great benefit.

For performing filtering in spatial domain, a mask should be moved on the image according to the following equation [10],

$$y(i,j) = \sum_{k=-a}^{a} \sum_{l=-b}^{b} h(k,l)x(i+k, j+l) \qquad (6)$$

Where, x and y are the input and output images, respectively. h is the mask that is going to be applied on the image. a and b are positive integers.

Theoretically, according to equation (6), any spatial filter can be implemented using the RNS. However, low pass filters such as median and mean ones include non-integer coefficients. This makes implementing them using the RNS a challenging task. Nevertheless, an alternative method to apply low pass filters and to overcome this obstacle is presented later in this section.

Furthermore, we have implemented a number of high pass filters, such as,

Sharpening filter: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$.

Laplacian filters as, $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

Roberts cross-gradient filters: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Other edge detection filters, where all coefficients

sum to 0: $\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$,

$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$, $\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$.

Since the coefficients of these filters are all integers, applying them using the RNS is straightforward.

As aforementioned before, applying low pass filters such as mean and median requires division operation, which is not a simple task in the RNS. Hence, to overcome this obstacle, we have used the following equation [11],

$$low\ pass = original - high\ pass \qquad (7)$$

Since subtraction is implemented simply and very efficiently in the RNS, using the above equation solves the challenge related to low pass filters.

Thus, implementing both low pass and high pass filters using the RNS becomes quite simple.

The proposed design has a similar structure to any RNS-based one. It consists of three main components; the first one is a ROM-based binary to residue converter that converts the binary pixels into their residue equivalents with respect to the moduli set {7, 15, 16}. The reason for choosing this moduli set is presented in the next section. Then, applying the filter based on equation (6) is done using three parallel modular channels according to the three moduli within the moduli set. Finally, the output residue values are converted back into their binary equivalents using a ROM-based residue to binary converter. The structure of the proposed design is illustrated in Fig. 1.

As can be noticed, the structures of both converters are ROM-based instead of combinational. The reason behind this is that, for this filtering application, the ROM-based converters turn out to be more efficient. The implementation results proved that these converters are faster by about 8.5% than those based on a pure combinational structure.

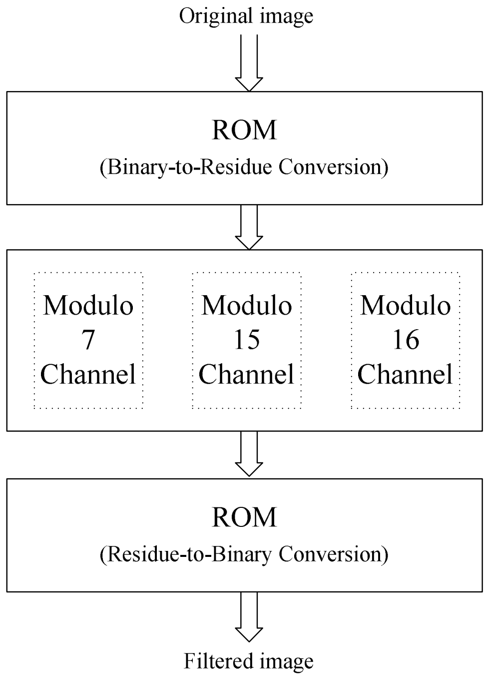


Fig. 1: The proposed design for implementing mask filters on a grayscale image

THE MODULI SET EFFECT ON THE FILTERING PROCESS

In image processing applications, the range of pixel values is [0,255]. However, since the filtering process is based on equation (6), the output filter can have a positive or a negative value. Nevertheless, in typical applications based on the binary number system, the output values are then scaled to the aforementioned range, i.e. if the output value is greater than 255 or less than zero, then it is scaled to 255 or to 0, respectively.

However, in the RNS, only the numbers within its dynamic range can be uniquely represented. The numbers beyond this range are then overlapped and inaccurate results are obtained. Hence, for such applications, an overflow detection unit is needed. However, this is not a simple task. Special components that represent additional overhead on the system are required. An alternative way to accomplish this task is via enlarging the dynamic range in such a way that it contains the possible output values (greater than 255 and less than 0).

In this section, the importance of using a proper moduli set with sufficient dynamic range is presented. Two moduli sets are compared to the one being used in our design. Both sets have been already used in image processing applications [5] and [6].

An example is illustrated in order to show the effects of using a moduli set with insufficient dynamic range. According to [9], the most efficient moduli set for applications that require medium dynamic ranges (less than 22-bits) is $\{2^{n-1} - 1, 2^n - 1, 2^n\}$. In this set, two moduli are of the form $(2^k - 1)$, which greatly simplifies the modular arithmetic units. We chose $n = 4$, thus, the used moduli set during our research is {7,15,16}. Its dynamic range = 1680 which is sufficient for image filtering application and eliminates the necessity to a special component for overflow detection. Many papers suggested using moduli sets with smaller dynamic ranges, such as {5,7,8} and {7,8,9} that provide $M = 280$ and $M = 504$, respectively [6], [5]. Since the possible pixel values in a digital image processing application have the range [0,255], these papers considered that using these sets would be sufficient. However, the following example clarifies the fact that this is not always true, except the case when using special components for overflow detection, which was not mentioned in any of those papers.

Example

Suppose the following pixel values in a part of a grayscale image,

91	94	142	183	200
113	146	186	194	211
176	187	177	207	106
190	184	221	112	36
185	207	122	39	45

Considering the outlined 3×3 neighbourhood of that image.

Suppose the following Laplacian filter that is going to be applied on that image,

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

1. <u>Using the standard weighted number system</u>

According to equation (6), the output filtered pixel value is,

$$\begin{bmatrix} 146 & 186 & 194 \\ 187 & 177 & 207 \\ 184 & 221 & 112 \end{bmatrix} \times \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} = -258$$

In standard image processing applications, negative numbers (in our case –258) are considered to be 0 (which refers to a black color).

2. Using the RNS based on the moduli set {5,7,8}

The dynamic range provided by this set is,

$\{5,7,8\} \Rightarrow M = 280$

Since the output of the used filter can be a negative value, the range should be split into two intervals in order to represent both negative and positive values. According to equation (3),

- The range of positive values is [0,139].
- The range of negative values is [140,279].

The above intervals clarify the fact that this dynamic range is not sufficient for such application (any filtered pixel greater than 139 will be corrected, according to equation (4), to a negative value, which is then scaled to 0 (a black color)).

Even though the intervals are not sufficient and back to our example, after forward converting those pixels with respect to the moduli set {5,7,8} and according to equation (6),

$$\begin{bmatrix} (1,6,2) & (1,4,2) & (4,5,2) \\ (2,5,3) & (2,2,1) & (2,4,7) \\ (4,2,0) & (1,4,5) & (2,0,0) \end{bmatrix} \times \begin{bmatrix} (1,1,1) & (3,5,6) & (1,1,1) \\ (3,5,6) & (4,4,4) & (3,5,6) \\ (1,1,1) & (3,5,6) & (1,1,1) \end{bmatrix} = \\ = (2,1,6) = 22$$

The number (22) belongs to the first interval, which indicates that it has a positive value and does not need any correction. Although the output value (22) refers to a dark color, it is not completely black as in the case of using the binary number system, where the output value is 0.

3. Using the RNS based on the moduli set {7,8,9}

The dynamic range provided by this set is,

$\{7,8,9\} \Rightarrow M = 504$

The ranges for representing negative and positive values are according to equation (3) as follows,

- The range of positive values is [0, 251].
- The range of negative values is [252,503].

Again, it is clear that this dynamic range is also not enough for such application.

For example, if an output filtered pixel is 253, which refers to an almost white color, then, according to the above intervals it will be corrected to a negative value, which is then dealt with as 0 (that indicates a black color).

Even though the intervals are not sufficient and back to our example, after forward converting the above 3×3 part of the image with respect to the moduli set {7,8,9} and according to equation (6),

$$\begin{bmatrix} (6,2,2) & (4,2,6) & (5,2,5) \\ (5,3,7) & (2,1,6) & (4,7,0) \\ (2,0,4) & (4,5,5) & (0,0,4) \end{bmatrix} \times \begin{bmatrix} (1,1,1) & (5,6,7) & (1,1,1) \\ (5,6,7) & (4,4,4) & (5,6,7) \\ (1,1,1) & (5,6,7) & (1,1,1) \end{bmatrix} = \\ = (1,6,3) = 246$$

The number (246) belongs to the first interval, which means that it has a positive value and does not need any correction. However, 246 refers to an almost white color. Again, the difference between the outputs of the RNS and binary number system is clear.

4. Using the RNS based on the moduli set {7,15,16}

The dynamic range provided by this set is,

$\{7,15,16\} \Rightarrow M = 1680$

The two intervals for representing both positive and negative values are,

- The range of positive values is [0,839].
- The range of negative values is [840,1679].

The sufficiency of both intervals is clear. It can be considered that the two intervals are more than enough for such application. However, we have used such big dynamic range to eliminate the necessity to an overflow detection unit, which represents an additional overhead on the overall performance.

Back to our example, after converting the pixels into their RNS representation, the filtering process is held according to equation (6),

$$\begin{bmatrix} (6,11,2) & (4,6,10) & (5,14,2) \\ (5,7,11) & (2,12,1) & (4,12,15) \\ (2,4,8) & (4,11,13) & (0,7,0) \end{bmatrix} \times \\ \times \begin{bmatrix} (1,1,1) & (5,13,14) & (1,1,1) \\ (5,13,14) & (4,4,4) & (5,13,14) \\ (1,1,1) & (5,13,14) & (1,1,1) \end{bmatrix} = (1,12,14) = 1422$$

$$1422 > M/2 \Rightarrow \textit{nagative value} \Rightarrow (1,12,14) = 1422 - M = -253$$

The number (1422) belongs to the second half of the dynamic range, which refers, based on equation (3), that it is a negative number.

After applying the necessary correction, we observe that the output value based on the moduli set {7,15,16} is the same as the one based on the binary number system (-253).

Hence, the importance of using a proper moduli set with sufficient dynamic range has been clarified.

IMPLEMENTATION RESULTS AND COMPARISONS

A 256×256 grayscale image was stored in a RAM, which was designed using Xilinx core generator V13.4. As aforementioned before, both forward and reverse converters were implemented as ROMs. In a similar manner, they were designed using Xilinx core generator. The rest of the blocks were described using VHDL and the whole design was compiled and implemented on Virtex4 XC4VLX15 FPGA device. Since the concept of the spatial filtering is the same, we have only applied Laplacian filter. The maximum frequency and power consumption were calculated using Xilinx Timing Analyzer and XPower Analyzer tools V13.4. The design goal and strategy was set to balanced.

Tab. 1 presents the maximum frequency and power consumption at clock frequency of 100 MHz after implementing the filter using the binary number system and the RNS based on the moduli set {7,15,16}. The superiority of the design based on the RNS is clear. The proposed design can operate at higher frequencies (up to 39.1%) and has less power consumption (by 23.7%).

Tab. 1: Comparison between implementing Laplacian filter on a grayscale image using the proposed design based on RNS and binary number system

	Binary number system	***RNS***	***Improved %***
Max. Freq. [MHz]	127.08	176.75	39.1%
PWR at 100 MHz [mW]	489	373	23.7%

Fig. 2 shows the results of applying sharpening and Laplacian filters using the binary number system and the RNS with three different moduli sets; {7,15,16}, {7,8,9} [5] and {5,7,8} [6]. The original input grayscale image is shown in Fig. 2. a. The output filtered image after applying the sharpening filter based on the standard binary number system is shown in Fig. 2. b. The output filtered images using the RNS based on the three moduli sets are show in Fig. 2 (c – e).

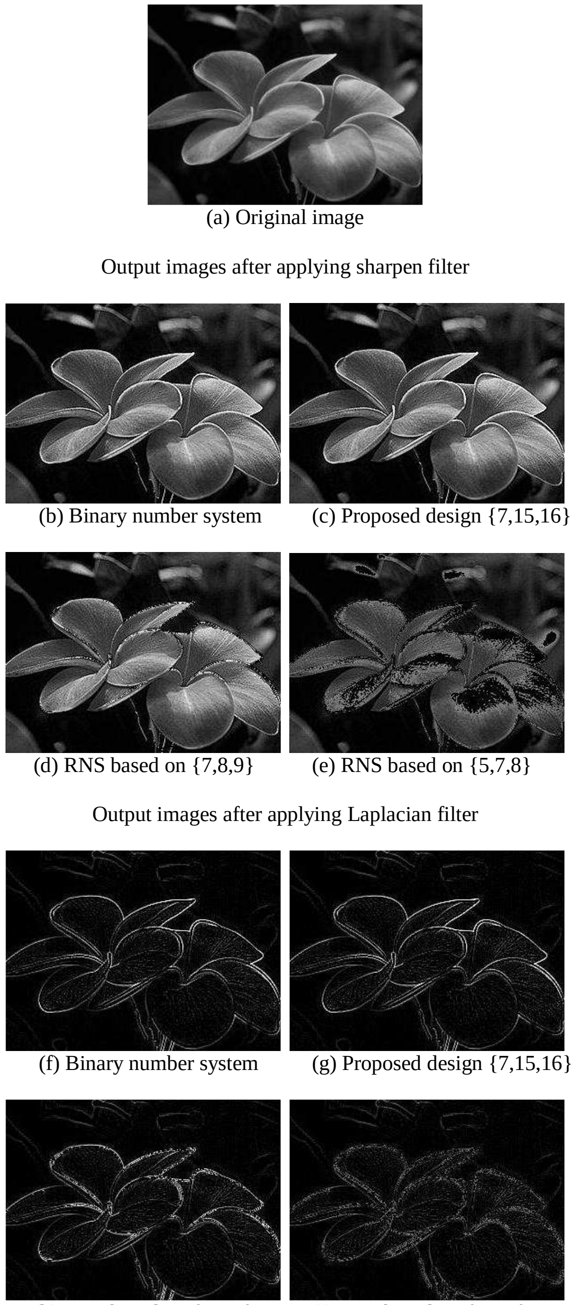
It is clear that the filtered images based on binary number system and on the proposed design are the same. However, the other two differ from the binary based one.

In a similar manner, the output image after applying Laplacian filter based on binary number system is shown in Fig. 2. f. The output images based on the RNS with different moduli sets are shown in Fig. 2 (g – i).

Again, the accuracy of the proposed design is clear. Its output image is identical to the one based on the binary number system, whereas the other two are not.

Hence, the negative effect of using a moduli set with insufficient dynamic range is clear. Fig. 2. (d, e, h and i) show the relatively distorted output images after applying the two filters using the RNS based on moduli sets {5,7,8} and {7,8,9}.

Different filtered images based on the proposed design have been compared to binary based ones. The outputs were 100% identical, which means that the proposed design based on moduli set {7,15,16} is not just fast and power reduced, but also accurate 100%.



(a) Original image

Output images after applying sharpen filter

(b) Binary number system (c) Proposed design {7,15,16}

(d) RNS based on {7,8,9} (e) RNS based on {5,7,8}

Output images after applying Laplacian filter

(f) Binary number system (g) Proposed design {7,15,16}

(h) RNS based on {7,8,9} (i) RNS based on {5,7,8}

Fig. 2: Comparison between output images after applying sharpening and Laplacian filters based on binary number system, the proposed design and other RNS based ones with the moduli sets {7,8,9} [5] and {5,7,8} [6].

CONCLUSIONS

An RNS-based image filtering design has been presented in this paper. All the calculations required for the filtering process are done using the RNS. The proposed design has been implemented on an FPGA and compared with its counterpart based on the standard binary number system. The implementation results proved that the proposed design can run at higher frequencies (up to 39.1%). Moreover it has less power consumption (by about 23.7%).

The second part of this paper was devoted for comparing the proposed design with other already published ones based on the accuracy of the filtering process. The images after filtering process based on the proposed design are identical to those based on the binary number system, whereas other RNS designs are not accurate. This means that these designs do not perform their task accurately contrary to the proposed one.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Industry and Trade of the Czech Republic under the MPO ČR č. FR-TI3/485 project and Prospective applications of new sensor technologies and circuits for processing of sensor signals project, No.FEKT-S-11-16.

REFERENCES

[1] A. Omondi and B. Premkumar, Residue Number Systems: Theory and Implementation, London, LDN, UK: Imperial College Press, 2007.

[2] M. Lu, Arithmetic and Logic in Computer Systems, New Jersey, NJ, USA: John Wiley & Sons, Inc., 2004.

[3] A. Nannarelli, M. Re, G.C. Cardarilli, “Tradeoffs between residue number system and traditional FIR filters,” in IEEE international symposium on circuits and systems, vol. 2, (Sydney), pp. 305–308, 2001.

[4] A. Ammar, A. Al Kabbany, M. Youssef, A. Amam, “A secure image coding scheme using residue number system,” in Eighteenth national radio science conference, vol. 2, (Mansoura), pp. 399–405, 2001.

[5] W. Wang, M.N.S. Swamy, M.O. Ahmad, “RNS application for digital image processing,” in 4th IEEE international workshop on system-on-chip for real-time applications, pp. 77–80, 2004.

[6] D.K. Taleshmekaeil, A. Mousavi, “The use of residue number system for improving the digital image processing,” in IEEE 10th international signal processing, (Beijing), pp. 775–780, 2010.

[7] D.K. Taleshmekaeil, H. Mohamamdzadeh, A. Mousavi, “Using residue number system for edge detection in digital images processing,” in IEEE 3rd international conference on communication software and networks, (Xi’an), pp. 249–253, 2011.

[8] S. Moharrami, D.K. Taleshmekaeil, “The application of the residue number system in digital image processing: propose a scheme of filtering in spatial domain,” Research journal of applied science, vol. 7, pp. 286–292, 2012.

[9] D. Younes, P. Steffan, “A comparative study on different moduli sets in residue number system,” in IEEE international conference on computer systems and industrial informatics, (Sharjah), pp.1–6, 2012.

[10] S.W. Smith, Digital Signal Processing: a Pracital Guide for Engineers and Scientists, New York, NY, USA: Newnes, 2003.

[11] A. McAndrew, An Introduction to Digital Image Processing with Matlab, internet book [online]. Place: School of Computer Science and Mathematics, MEL AU, 2004 [cit. 2013-07-23]. Available from WWW: <http://visl.technion.ac.il/>.