Computational Complexity Evaluation of Reversible Integer to Integer Wavelet Transforms Used in IoT Systems

Dr. Kesavan Gopal[1], G. Kalyani[2], G. Lok Prakash Raju[3], G. Revanthreddy[4], P. Koteswara Rao[5], V. Prudhvi Raj Yadav[6]

[1] Professor, School of Electronics and Electrical Engineering (SEEE), Lovely Professional University, Phagwara, Punjab, India.

[2, 3,4,5,6] Students, B. Tech Final Year, SEEE Lovely Professional University, Phagwara, Punjab, India.

Abstract: Internet of Things (IoT) is an emerging area of research primarily focusing on automation, resulting in cost benefits. IoT and embedded systems deployed in the field generally has limited processing and battery capability. However, for transmission of video, image and audio data over IoT communication environment, requires heavy processing capability at the edge nodes and need to consume least power to sustain the working of systems. To achieve the system efficiency, either the processing capability of the edge node is to be improved or the processing algorithm complexity is to be reduced. For IoT based embedded system design, predicting and estimating the complexity of the computations involved in processing is very important in order to implement the system. Many of the image, video and audio processing standards uses wavelet transform as its core for energy compaction of data, hence it is important to do the computational complexity analysis of such transforms. Integer-to-Integer wavelet transforms is the widely used method for applications requiring reversibility like lossless compression. In this paper, computational complexity of various integer-to-integer wavelet transforms namely (1) 2/6 Transform (2) 2/10 Transform (3) SPB Transform (4) SPC Transform are calculated, implemented and analyzed.

Index Terms **- IoT Systems, Integer-to-Integer Wavelet Transforms, Computational Complexity.**

I. INTRODUCTION

Internet of Things (IoT) is the new area of research and development in recent years. By definition, IoT is a device connected to internet. The general architecture of IoT system consists of edge nodes, network gateways, data processing modules and application management units. The focus of this paper is mainly on the edge nodes and its pros and cons in IoT applications. The edge node falls under the sensing layer of IoT architecture and it is made up of sensors and actuators. These devices accept the incoming physical parameters, processes it in the first level before being forwarding it to the gateways for analytics and transmission. The kind of physical parameters includes, temperature & humidity, pressure, light, optical, gas & smoke, proximity, water quality, chemical, infrared, level, acceleration, angular rotation (gyroscope), motion detection and image sensing. The scope of this paper is further narrowed down to imaging sensors in IoT environment. Image sensors are a kind of transducers, where the incoming optical images are converted into an equivalent electrical signal for processing. These imaging sensors are used in CCTV cameras for surveillance, IRIS and retina scanning, medical imaging of patient in clinics, night vision & thermal imaging for surroundings monitoring and security etc. All of these areas are either one or the other ways are moved towards the direction of IoT applications. The main drawback of edge nodes in IoT are its least processing capability [6] and demand for high energy efficiency. In order to meet this demand, the imaging sensors connected to the edge nodes need to be high processing capacity and extremely energy efficient. To achieve the improved processing capability, the algorithm used in image, video processing is to be simplified. Any image or video processing standards uses various transforms for its energy compaction and data reduction. One among the widely used transform is Wavelet Transform (WT) and it is the core part of any video and image processing in general. The computational complexity of the WT used in edge nodes [7] is to be understood in order to optimize and to yield the reduction in the number of operations like additions, subtractions, shifting and multiplications used in computing the transform Further the paper is organized as follows: section 2 describes an overview of Wavelet Transforms, Section 3 describes various ITI wavelet transforms, section 4 describes the computational Complexity of ITI transforms, section 5 describes the implementation and analysis of ITI wavelet transforms, section 6 concludes the paper.

II. WAVELET TRANSFORMS – AN OVERVIEW

Fourier Transform is one of the widely used methods of transforming a time domain signal into its frequency domain [1][3][4]. There are certain applications which require information details not only in the time domain but also in the time-frequency domain for better analysis and meaningful interpretation of signal. With the advent of Short-Term Fourier Transform (STFT), analysis in time-frequency domain is made possible. Then comes the Wavelet Transform and it is a transform which is used to transform the information in time domain into the frequency domain and helps to carry out the multi-resolution analysis [12] of the given signal in time-frequency domain. It provides the information about the presence of frequency in the signal and at what instance it exists. The difference between the Fourier Transform analysis and Wavelet transform based analysis is that, the former uses the sines and cosines for analysis and the later uses the wavelets for analysis. By definition, wavelet transform is nothing but a representation of a function by wavelets; It is classified as Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT) [4]. In this work, the focus is towards the DWT. DWT's are a widely used transform in image and video coding, due to the reason that, it provides the perfect energy compaction of the 2D signals. There are various wavelets families that are presently used for signals analysis and it includes Haar, Coiflets, Daubechies, Morlet, Biorthogonal, Meyer, and Symlets Mexican Hat etc... Based on the results of computation of the wavelet transform, it is classified as floating-point wavelet transforms and integer-to-integer wavelet transforms (ITIWT) [10]. These ITIWT are primarily suitable for usage in reversible watermarking [7][8] as well as in lossless image and video coding applications. The computational complexity of these transforms plays a vital role in selecting the transform and its hardware implementation [9]. Many applications in video and image coding requires the wavelet transform to be lossless, means that the 2-D pixel are converted into equivalent integers in transform domain. The ITIWT, its properties, transform and complexity computations are described in the subsequent section.

III. ITI WAVELET TRANSFORMS

In image and video coding lossless performance and its computational complexity determines the overall system performance [2]. Many of the image and video compression [11] and watermarking application requires reversible in nature for which it is necessary to use the reversible transforms, like that used in JPEG2000 and DICOM, Wavelet Video CODEC etc. There is various reversible integer-to-integer transform are available namely 5/3 [9], 2/6, SPB, 9/7-M, 2/10, 5/11-C, 5/11-A, 6/14, SPC, 13/7-T, 13/7-C, 9/7-F etc. The performance of these integer-to-integer transforms are highly comparable with that of its real irreversible transform counterpart and outperforms the requirements of lossless CODEC in various applications. The performance metrics of wavelet transform used in the evaluation includes the low computational complexity, memory usage, look-up-table based storage and time to compute the coefficient. These metrics are better understood, then the IoT consisting of very least computational performance can be designed efficiently and make it cost effective. The transforms mentioned as above are1-D in nature and extendible to 2-D. The transform results can also be obtained by using the analysis and synthesis filter bank methods collectively called as filter bank methods. For the purpose of implementation one set of ITIWT under linear phase FIR filters category, namely 2/6 and 2/10 transforms, and another set of ITIWT under non-linear phase filter banks, namely SPB, SPC are considered. The implementation of other transforms is more or less exactly the same as that of implementation of these transforms. Since the transform under consideration are 1-D in nature, two-dimensional image pixels can be transformed by applying the low and high pass filters in row wise, then the row wise result is subjected to the same low and high pass filters in column wise, yields 2-D frequency domain coefficients. The order of performing the forward direction like row-column wise need to follow in the reverse direction like column-row wise. Unless otherwise this order is followed the reversibility from frequency domain coefficient to the time domain pixel is not guaranteed. The following notations are used for in the computations of transform: x[n]-input signal; s[n] –low pass sub band signal; d[n] – High-pass sub band signal. The forward transforms of 2/6, 2/10, SPB and SPC are mentioned in (1) (2) (3) (4) respectively. The inverse transforms are equivalently derived from the forward transforms.

$$\text{2/6 Transform} = \begin{cases} d_1[n] = d_0[n] - s_0[n] \\ s[n] = s_0[n] + \lfloor \frac{1}{2} d_1[n] \rfloor \\ d[n] = d1[n] + \lfloor \frac{1}{4}(-s[n+1] + s[n-1]) + \frac{1}{2} \rfloor \end{cases} \quad (1)$$

$$\text{2/10 Transform} = \begin{cases} d_1[n] = d_0[n] - s_0[n] \\ s[n] = s_0[n] + \lfloor \frac{1}{2} d_1[n] \rfloor \\ d[n] = d_1[n] + \lfloor \frac{1}{64}(22(s[n-1] - s[n+1]) + 3(s[n+2] - s[n-2])) + \frac{1}{2} \rfloor \end{cases} \quad (2)$$

$$\text{SPB Transform} = \begin{cases} d_1[n] = d_0[n] - s_0[n] \\ s[n] = s_0[n] + \lfloor \frac{1}{2} d_1[n] \rfloor \\ d[\mathrm{n}] = d_1[n] + \lfloor \frac{1}{8}(-3s[n+1] + s[n] + 2s[n-1] + 2d1[n+1]) + \frac{1}{2} \rfloor \end{cases} \quad (3)$$

$$\text{SPC Transform} = \begin{cases} d_1[n] = d_0[n] - s_0[n] \\ s[n] = s_0[n] + \lfloor \frac{1}{2} d_1[n] \rfloor \\ d[n] = d_1[n] + \lfloor \frac{1}{16}(-8\mathrm{s}[n+1] + 4s[n] + 5s[n-1] - s[n-2] + 6d1[n+1]) + \frac{1}{2} \rfloor \end{cases} \quad (4)$$

IV. COMPUTATIONAL COMPLEXITY OF ITI WAVELET TRANSFORMS

The four ITIWT mentioned in previous section can be computed using integers of fixed-point arithmetic. In order to compute the transform, the arithmetic operators required are adders, subtractors, multipliers and dividers [2][5]. Multiplication and division operations can be implemented by bit shift operations. For the purpose of transform computation in 1D, the number of adders, multipliers and shifters required per pair of input signal is given in Table 1.0.

Table 1.0

Transforms	Adders	Shifters	Multipliers	Total
2/6	5	2	0	7
2/10	7 (10)	2 (6)	2 (0)	11
SPB	7 (8)	4 (3)	1 (0)	12 (11)
SPC	8 (10)	4 (5)	2 (0)	14 (15)

In the above table, first numbers are the outcome of the direct implementation, the second one present inside the brackets are the one obtained by replacing the multiplication operation equivalently with the addition and shifting. These parameters are of high importance when the hardware and software implementation are carried out, since the multiplications are costlier than the adders and shifters. Closely looking at the table 1.0 the number of computations with each other is close together same. Whenever there is an implementation of Wavelet Transforms, integer to integer wavelet transforms is preferred over their real counterpart due to simpler implementations, including the number of bits required to store the processed result. For example, to store the intermediate results, the transform requires 32-bit wide memory and it can be brought down to 16-bits wide by adopting dynamic range of allocation. Even further smaller number of memory elements can be used for the transform coefficients, by knowing dynamic values of the various regions like Low-Low (LL), Low-High (LH), High-Low (HL), and HH (High-High). It clearly goes to show that, ITIWT uses reduced computational complexity and memory word requirements. Finally, the overflow and underflow values of transform coefficients are quite lesser or nearly nil in case of ITIWT transform in comparison to the real-lossy transforms available, causing the artifacts resulting in the distorted image after reconstruction. A detail about the implementation and analysis of four transforms namely 2/6, 2/10, SPB, SPC is presented in the subsequent section.

V. IMPLEMENTATION AND ANALYSIS OF ITI WAVELET TRANSFORMS

The methodology adopted in verifying the functionality and to evaluate the computation complexity of four integer-to-integer wavelet transforms is implemented in visual Studio 2019 compiler using C language. The sequences of input pixels are stored in the text files and it is read and copied into the two-dimensional array of 4 x 4 matrix. This array is once again copied into 16 variables suitable for easy hardware implementation. When moving from software implementation to the hardware implementation suitable for IoT systems, these variables are replaced with the appropriate memory addresses. For corner pixels either symmetric or boundary extension of pixels are carried out to overcome the loss of reversibility. Initially the pixel values are low pass filtered in row wise, then the resultant 1D coefficient are low pass and high pass filtered in column wise in order to get the 2D coefficient [9]. The computed coefficients are stored back in the memory with varying bit depth levels. These stored 2D coefficients are suitable to be used for further image encoding techniques. In a similar way, the forward transforms are slightly modified and the reverse integer-to-integer wavelet transforms too is computed. Thus, the resultant pixel values computed after the application of reversible transform is exactly same as that of the input pixels used for the forward transform bit by bit computationally, confirming the reversibility. The pseudo code of integer-to-integer wavelet transform is presented as follows:

Pseudo code of ITI Wavelet transforms

```
Step 1: Start
Step 2: Read in the input pixel matrix.
Step 3: Initialize the variables row, column, length, adders, shifters and multipliers with zero
Step 4: Align the pixel matrix and compute the extension of pixels at boundaries
Step 5: for row in 0 to length-1
            for column in 0 to length-1
                compute the low pass filter bank coefficients
                   compute the high pass filter bank coefficients
                        Compute the number of adders, shifter, and multipliers
            end loop
        end loop
Step 6:  Repeat steps 1 to 5 for all transforms
Step 7: end
```

V. CONCLUSION

In this paper, an evaluation of computational complexity of four integer-to-integer wavelet transforms namely (1) 2/6 Transform (2) 2/10 Transform (3) SPB Transform (4) SPC Transform is carried out successfully. It is being observed that the 2/6 transform uses 2 adders, 5-shifters and 0-multipliers totalling a complexity of 7; the 2/10 transform uses 7- adders, 2-shifters, and 2-multipliers totalling a complexity of 11 when implemented directly, and requires 10-adders, 6-shifters totaling a complexity of 16 when implemented without multipliers; the SPB transform uses 7-adders, 4-shifters, and 1-multiplier totalling a complexity of 12 when implemented straight forward, and requires 8-adders, 3-shifters totalling a complexity of 11; finally the SPC transform uses 8-adders, 4-shifters, 2-multipliers totalling a complexity of 14 when implemented directly and 10-adders and 5-shifters when implemented without multipliers. As seen from these computational complexities of various transforms, a transform whose filter bank uses computations in the power of two., yields better figures in terms of adders and shifter, on the other way the transform which uses the multiplier attains the higher number of computational complexity figures. Therefore, while designing IoT edge nodes processing an image or video, requires a priori knowledge about the computational complexity for the best system performance.

REFERENCES

[1] Prochazka, A., Kngsbury, N., Payner, P.J.W., Uhlir, J., Signal Analysis and Prediction, Chapter-2, pp 27-46, ISBN 978-1-4612-7273-1

[2] Michael David Adams, Ph.D. Thesis titled, Reversible integer-to-integer wavelet transforms for image coding, pp 86-107, chapter 5. Sep 2002, accessed at https://www.ece.uvic.ca/~frodo/publications/phdthesis.pdf.

[3]Amara Graps, An Introduction to Wavelets, pp. 1-18, accessed at https://www.eecis.udel.edu/~amer/CISC651/IEEEwavelet.pdf

[4]Elin Johanson**,** Wavelet Theory and some of its Applications, Licentiate Thesis, pp 1-90, accessed at https://core.ac.uk/download/pdf/22879255.pdf

[5]M.D. Adams F. Kossentni, Reversible Integer-to-Integer Wavelet transforms For Image Compression: Performance Evaluation and Analysis, IEEE Transactions On Image Processing, Vol.9, Issue.6, June 2000, pp 1010-1024.

[6]Huifeng Wu, Junjie Hu, Jiexiang Sun and Danfeng Sun, Research Article titled, Edge Computing in an IoT Base Station System: Reprogramming and Real-Time Tasks, pp 1-10, Vol. 2019, Article ID: 4027638, accessed at https:/doi.org/10.1155/2019/4027638.

[7]Kevin Fogarty and ED Sperling, Blog Article, Edge Complexity to Grow for 5G, accessed at https://semiengineering.com/edge-complexity-to-grow-for-5g/

[8]Zhengwei Zhang, Mingjian Zhang and Liuyang Wang, Research Article, Reversible Image Watermarking Algorithm based on Quadratic Difference Expansion, Vol. 2020, Article ID 1806024, https://doi.org/10.1155/2020/1806024

[9]Maria E. Angelopoulou and Peter Y.K. Cheung Konstantinos Masselos, Yiannis Andreopoulos, Implementation and Comparison of the 5/3 lifitng 2D Discrete Wavelet Transform Computation Schedules on FPGAs, Journal of VLSI Processing 2007, DOI:10.1007/s11265-007-0139-5.

[10] HAO Pengwei, SHI Qingyun, Invertible Linear Tranforms Implemented by Integer Mapping, Science in China, Series E, April 2000, Vol.30, pp 132-141.

[11] M. J. Gormish, E. L. Schwartz, A. F. Keith, M. P. Boliek, and A. Zandi. Lossless and nearly lossless compression of high-quality images. In *Proc. of SPIE*, volume 3025, pages 62–70, San Jose, CA, USA, March 1997.

[12] A. Said and W. A. Pearlman. An Image Multiresolution Representation For Lossless and Lossy Compression. *IEEETrans. On Image Processing*, 5(9):1303–1310, September 1996.