International Symposium on Signal Processing and its Applications (ISSPA), Kuala Lumpur, Malaysia, 13 – 16 August, 2001. Organized by the Dept. of Microelectronics and Computer Engineering, UTM, Malaysia and Signal Processing Research Centre, QUT, Australia

IMPLEMENTATION OF WAVELET CODEC BY USING TEXAS INSTRUMENTS DSP TMS320C6701 EVM BOARD

Zulkalnain Mohd Yusof, Nurulfajar Abd Manap, Ishak Suleiman, Sheikh Hussain Sheikh Saleh, Zulfakar Aspar.
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
Johor Bahru, Johor.

ABSTRACT

This paper describes the implementation of the Wavelet Codec (Encoder and Decoder) by using the Texas Instruments DSP (Digital Signal Processor) TMS320C6701 on the EVM (Evaluation Module) Board. The Wavelet Codec is used to compress and decompress gray scale image for real time data compression. The Wavelet Codec algorithm has been transferred into C and assembly code in Code Composer Studio in order to program the 'C6xx DSP processor. The capability of the 'C6xx to change the code easily, correct or update applications, reducing the development time, cost and power consumption. With the development tools provided for the 'C6xx DSP platform, it created easy-to-use environment that optimizes the devices' performance and minimizes technical barriers to software and hardware design.

I. INTRODUCTION

Recently, there are so many modern digital technology data and applications such as high production video games, high definition television (HTDV), digital video conferencing, video telephony, medical imaging or virtual reality have becoming extremely high bit rate and storage intensive. Simultaneously, with the advents of high-speed networks (ISDN, ATM) and the rise of multimedia industry, the application of image and video compression technology becomes increasingly important. The raw digital image and video signal usually contain huge amount of information and therefore require a large channel or storage capacity. In spite of advances in communications channel and storage capacity, the implementation cost often put constraint on capacity. Generally, the transmission or storage cost increase with increase in bandwidth requirement. To meet the channel or storage capacity requirement, it is necessary to employ compression techniques, which reduce the data rate while maintaining the subjective quality of the decoded image or video signal. Image compression techniques exploit statistical redundancies in the data and human visual system to achieve compression. While still image coding exploits only spatial redundancies, video coding exploits both spatial and temporal redundancies to achieve high compression ratios.

Due to the increasing need of video compression for storage and transmission, many international standards have emerged. Different standards exist for different applications. For instance, the JPEG standard has been designed for coding still images, whereas the H.261 standard, which is part of H.320 standard, is designed for video-conferencing, and the MPEG1, MPEG2 and MPEG4 standards are designed to support more advanced video communication applications. These image and video standards however are block-transform based coders and are known to suffer from blocking effect especially at low bit rate [1,2]. Recently, wavelet based coding schemes have demonstrated as a promising candidate for compressing both still image and video sequence at low bit rate. The wavelet transform has become a cutting-edge technology in image compression research.

Usually, the compression algorithms are implemented by using digital hardware to perform specific functions in the algorithm. Hardware implementation of video compression algorithm can provide the speed and density but lacks the flexibility. However, recent advances in semiconductor technology have improved the performance and reduced the price of general purpose DSP. This makes general purpose DSP an attractive solution for implementing video compression algorithm especially for low to medium video frame rate applications. The major advantage of using DSP is programmability, speed performance, and fast development cycle. On the other hand, DSP is inherently stable, reliable and repeatable. Development can be done in high level language and critical code can be hand coded in assembly language to further increase speed of execution. The Texas Instruments TI TMS320C6000 ('C6xx/'C67xx) are the latest and highest performance fixed and floating point DSP processors available from TI. These processors are 10 times faster than any other processor on the market at the present time, which are 2400 million instructions per second (MIPS) at 300 MHz [4]. The 'C6xx are the first processors to benefit from highly efficient C compiler and assembly optimizer.

There are two types of programmable DSP: fixed point DSP and floating point DSP. Fixed point DSP is cheaper, faster and is suitable for large volume production. However, translating algorithms into fixed point assembly code is much more challenging because of the needs to handle scaling of data at each step of the arithmetic computations to prevent overflows, while maintaining the accuracy. Floating point DSP is more expensive but algorithms mapping to assembly code is relatively easier and quite straightforward, which can lead to faster design cycle.

This paper describes the implementation of the Wavelet Codec on the 32 bit floating point TI DSP processor TMS320C6701 on the EVM board. The organization of the paper is as follows: Section II provides a brief description of the Wavelet Codec image compression algorithm and the main components used. Section III provides information on the DSP implementation. And finally section IV concludes the paper.

II. WAVELET CODEC DESCRIPTION

This section presents a brief description of the wavelet codec for image compression algorithm. For detailed description on wavelet transform, reader is referred to [6,7]. Figure 1 shows the block diagram of the wavelet codec. It comprises three main stages: transformation (wavelet decomposition), quantization and coding. The image is decomposed using 4 level wavelet filters, which result in 13 subbands. Each subband is then quantized using a step size value, which is used to control the total bit rate. The quantizer stage is followed by runlength and Huffman coding. The inverse process or decompression is done by following the reverse order.

Transformation

The principle of wavelet image coding is based on the decomposition of an image into a number of frequency bands referred to as subbands. In order to increase efficiency, quantizer and coder for each band can then be designed to match the statistic and energy activity of each subband. Coding efficiency can be further improved by taking advantage of inter and intra subband dependencies. This paper will be concentrated on intra subband only.

In Figure 2(a), Lena image was decomposed into four subbands called LL, LH, HL and HH. Then, Figure 2(b) shows the LL band decomposed into four subbands called LL_2, LH_2, HL_2 and HH_2. In this research work, the original image will be decomposed into 4 levels of 13 subbands, as shown in Figure 2(c). According to H. J. Barnard, the decomposition using tree splitting where only the low-pass subbands are split up to depth of 4 would be better for coding purposes.

Quantization

Each subbands are quantized using uniform scalar quantizer, which is used to control the total bit rate. Each subbands are quantized using the step size defined by the quantization matrix and is implemented for the frame of image, followed by rounding to the nearest integer

$$F_q(u,v) = Round\_Integer\left[\frac{F(u,v)}{Step(u,v)}\right]$$

The quantization is performed by the following operation:

$$F_{rec}(u,v) = F(u,v) * Step(u,v)$$

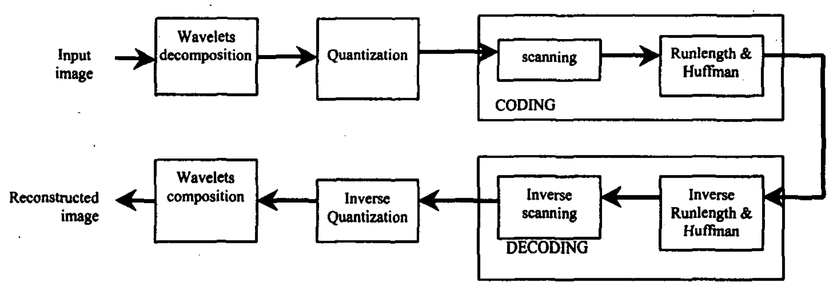

Figure 1: Block diagram of Wavelet Codec

International Symposium on Signal Processing and its Applications (ISSPA), Kuala Lumpur, Malaysia, 13 – 16 August, 2001. Organized by the Dept. of Microelectronics and Computer Engineering, UTM, Malaysia and Signal Processing Research Centre, QUT, Australia

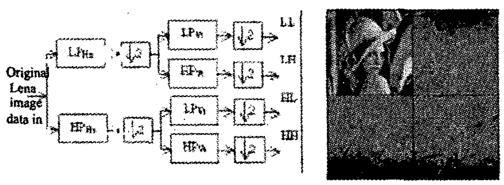

Figure 2(a): Decomposition Lena image into four subbands.

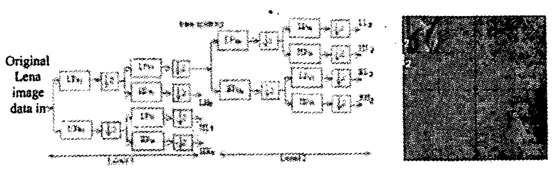

Figure 2(b): The LL band of Lena image decomposed into four subbands.

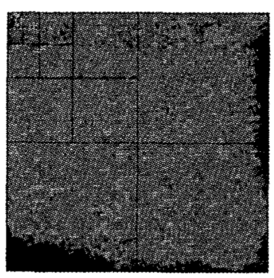

Figure 2(c): Lena image decomposed into 4 level subbands.

Coding

The quantizer stage will be followed by the scanning process, zero run length coding and Huffman coding. The performance of Wavelets can be improved by exploiting the cross-correlations among the subbands in the encoding process. The idea behind scanning process is to get the maximum energy compactness in order to get maximum compression ratio. During the scanning process the wavelet coefficient will be compacted according to the sequence proposed by Shapiro. The process is repeated from beginning to end among the subbands, which have the correlations between them, until all coefficients are being scanned. This rearrangement is needed because the lower frequency coefficient, which is sensitive to human visual system, is located at the level 4 LL band of subband. While the higher frequency coefficient, which is less sensitive to human visual system is located at the level 1 HH band [1,2].

Human visual system is less sensitive at higher frequencies. These higher frequency components can be coarsely quantized than the lower frequency components, with little effects on image quality. The courser quantization of high frequency coefficient results in many zero-valued coefficients. Rearranging the subbands in the scanning process leads to a long string of zeroes. The long string of zeroes can be compactly or efficiently represented using zero run-length coding. The coefficients are then coded by using Huffman coding.

To reconstruct the image, the decoder basically performs the three main inverse operations in reverse order as shown in Figure 1.

III. WAVELET CODEC DSP IMPLEMENTATION

The wavelet codec was implemented on a TMS320C6701 (EVM board from TI). The wavelet codec software was written in C and assembly code, which allows the code to be changed, updated and modified easily by using Code Composer Studio. To aid the development, the entire algorithm was first simulated on general-purpose computer using the C language, in Microsoft Visual C++. Each components of the Wavelet codec is then converted to C and assembly language that is compatible with Code Composer Studio environment. Even though the code have been simulated by using Microsoft Visual C++, the validity of the entire program was rechecked by using debugger and profile simulator platform provided by Texas Instruments for 'C6xxx DSP processor generation. The code was written for the 'C6701 processor and provides all initializations required on the DSP side, routines for communication with the host and the functions of wavelets block.

A host computer is used to store and read/display the compressed and uncompressed image respectively from the TMS320C6701 system. The communication between the host and TMS320C31 system is via the PCI expansion slot inside a PC. A C/C++ program was written to run on the host for the purpose of downloading the code/data, initiating the execution, reading and displaying the reconstructed image.

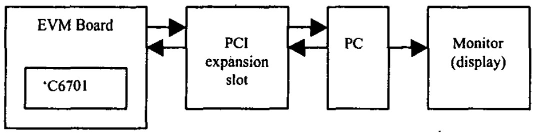

Figure 4: Wavelet Codec implementation by using TMS320C6701 DSP EVM board.

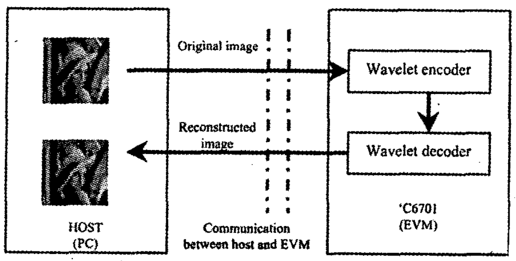

Figure 5: Testing environment of Wavelet Codec .

V. RESULT

The Wavelet codec by using 'C6701 was found to work. The original Lena image was compressed and decompressed by using wavelet codec algorithm. The reconstructed Lena image was compared with the original image, in order to analyze the quality of the reconstructed image, as shown in Figure 6(a) and 6(b). Table 1 show the performance of wavelet compression components by using the profile statistic of Code Composer Studio. The cycles can be slightly reduce by writing the C code into assembly, which is in progress.

Wavelet components module	Number of cycles	
	Build without optimization	Build with file optimization.
Forward DWT	53,776	47,178
Inverse DWT	26,538	24,517
Quantization	19,449	3317
Inverse Q	1806	104
Scanning	14074	3558
Zero Run Length	4834	3627
Inverse ZRL	2592	1612
Huffman Encode	15,213	1537
Huffman Decode	3068	1755

Table 1: Number of cycles for Wavelet codec components by using profile statistic Code Composer Studio.

VI. CONCLUSION

The Wavelet codec was successfully implemented on floating point TMS320C6701. The program was written in C and assembly language to achieve fast program execution. The capability of the 'C6xx to change the code easily and update applications have reducing the development time and cost. Besides, with the development tools provided for the 'C6xx DSP platform, it created easy-to-use environment that optimizes the devices' performance and minimizes technical barriers to software and hardware design.

Figure 6(a): Original Lena Image

Figure 6(b): Reconstructed Lena Image

REFERENCES

[1] J.M Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", IEEE Trans. Signal Processing, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[2] J.M. Shapiro, "Smart Compression Using the Embedded Zerotree Wavelet (EZW) Algorithm", In Proc. of the 27th Asilomar Conf. on Signals, Systems and Computers, vol. 1, pages 486-490, November 1993.

[3] Texas Instrument, "TMS320C6000: Programmer's Guide", Manual. Dallas, USA, 2000.

[4] Naim Dahnoun, "DSP Implementation using the TMS320C6000 DSP Platform", Prentice Hall. London, 2000.

[5] Zulkalnain Mohd Yusof, et. al., "Real time implementation of baseline JPEG decoder using floating point DSP TMS320C31", IEEE 2000 TENCON Proceedings, vol. 3, pages 383-386, September 2000.

[6] Amara Graps, "An Introduction to Wavelets". IEEE Computer Science and Engineering. Volume 2, Number 2. USA, 1995.

[7] Y. T., Chan, "Wavelet Basics." Kluwer Academic Publishers. Norwell, MA, 1995.