

A Hyper-Domain Image Watermarking Method based on Macro Edge Block and Wavelet Transform for Digital Signal Processor

Yi-Pin Hsu and Shin-Yu Lin

Abstract—In order to protect original data, watermarking is first consideration direction for digital information copyright. In addition, to achieve high quality image, the algorithm maybe can not run on embedded system because the computation is very complexity. However, almost nowadays algorithms need to build on consumer production because integrator circuit has a huge progress and cheap price. In this paper, we propose a novel algorithm which efficient inserts watermarking on digital image and very easy to implement on digital signal processor. In further, we select a general and cheap digital signal processor which is made by analog device company to fit consumer application. The experimental results show that the image quality by watermarking insertion can achieve 46 dB can be accepted in human vision and can real-time execute on digital signal processor.

Keywords—watermarking, digital signal processor, embedded system

I. INTRODUCTION

BY internet progress, the copyright topic becomes an essential issue. From signal point to expend to multimedia application such as image, audio and video; user need an efficient method to protect them authority, therefore watermarking is appropriate method. Digital watermarking has emerged as a potentially effective tool for multimedia copyright protection, authentication and tamper proofing in [1]. Watermarking is the process of inserting hidden mark in an image by introducing modifications to its pixels with minimum perceptual disturbance. In [2], proposed a robust method to against manipulation. Even though the method is quite robust, the original image must be present for watermarking recovery. Recently, independent scheme on original image become main research direction. In past approach, a general ideal is to locate watermarking on frequency domain. Ruanaidh *et al* [3] first proposed a watermarking scheme based on transform invariants via applying Fourier-Mellin transformation to the magnitude spectrum of an original image. However, the result of stego-image quality is poor due to interpolation errors in [4]. In [5-6], the watermarking will be embedded on Discrete

Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) domain individually. The watermarking can be identified by calculating the correlation between watermarking sequence and the coefficients of the watermarked image. However, illegal correlation status can not embed the watermarking and needs more computation to find a good location. The well-known patchwork watermarking in [7-8] inserted a message by supporting that two sets of randomly selected pixels are Gaussian distribution with zero mean. The embedded method is by shifting the mean values between groups of two sets of pixels. Recently, in [9], a template-based patchwork watermarking for color image was proposed. The method is focus on YUV (luminance and chroma) color space and only fit for color image. The Y and V spaces are extracted robust feature and classify U into many blocks for watermarking in reference to the robust features extracted. In another method, the watermarking by histogram specification is proposed. In [10-12] the watermarking is a predefined histogram; by referring to the predefined histogram the pixels in the original image are regrouped to generate the watermarked image which has the same shape of histogram as the watermarking has. The histogram can also be exploited as the reference for reversible watermarking in [13-14], the histogram is used to seek possible redundant information for embedding bits as much as possible. Beside, an important assumption that there is no any distortion on the marked image for reversible watermarking is needed.

Although above description has good performance to embed watermarking and keep high Peak signal noise ration (PSNR) in original image, the system will be requested a lot of memory to store data and fast processing unit to calculate in real time application thus the method is unsuitable.

In this paper, we proposed a novel composite scheme which includes macro edge properties for temporal operation and DWT transform for spatial operation. In further, self-similarity characteristic of watermarking which can be performed by sub-sampling also be used as another protection layer and become two the same size watermarking to embed on image.

The rest of this paper is organized as follows. In section II, we introduce the digital signal processor (DSP) which calls as Blackfin561. Section III describes the watermarking embedding algorithm. An overall flow combine algorithm and how to implement algorithm on DSP system are depicted in

Yi-Pin Hsu is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan. (Corresponding author to provide phone: +886-9-39902564; e-mail: hsuyipin@gmail.com).

Shin-Yu Lin is also the Department of Electrical and control Engineering, National Chiao Tung University, Hsinchu, Taiwan.

Section IV. The section V describes a lot of balance comparison by experimental results. A conclusion is drawn in last section.

II. DSP SYSTEM OVERVIEW

In order to improve the value and feasibility, a real time embedded system application become a standard verification. Although Application Specific Integrated Circuit (ASIC) is low price and high performance solution, its flexibility is a drawback for multi-function integration. A powerful and low price DSP is essential consideration for usage of complex computation. In market, a lot of companies provide different solutions. Texas Instrument (TI) focuses on high performance direction such as advanced video coding. Although the performance is very excellence, the price is great loading in consumer application. Fortunately, Analogy Device Company [15] provides another solution which emphasize on balance between performance and price. The Blackfin processor focuses on consumer multimedia applications in the series of DSP family of Analogy Device Company. Especially, the ADSP-BF561 is a member of the Blackfin processor which the heart of this device are two independent enhanced processor cores to offer high performance and low power consumption. The core architecture combines a dual-MAC signal processing engine which belongs to signal-instruction multiple-data (SIMD) structure. The SIMD can efficient operate data in lower cycles. In power management, the Blackfin products provide dynamic management solution which can detect operation frequency and voltage to slight tune power supply. Beside another important issue is memory arrangement, hierarchical architecture such Level 1 (L1) and Level 2 (L1) as caches are implemented for low memory switch. Generally, L1 can offer zero memory waiting time is close DSP core and L2 need more cycles to move data in comparing with L1. Comparatively, L2 can own large capacities compare with L1 but lower system cycles than external memory. Due to Blackfin561 is dual core, the memory map has some different. Each core include own L1 independently, and L2 is shared memory for both. The overview of memory allocation can be depicted as follows.

- Level 1 memory size: 100 KB (include data and instruction)
- Level 2 memory size: 128 KB
- External memory size: can support to up to 512 MB of SDRAM by PC133-compliant controller.

Finally, direct memory access (DMA) potential and channels are another bolster point. ADSP-BF561 has three independent DMA controllers such as DMA1, DMA2 and IDMA. Each of DMA1 and DMA2 has twelve peripheral DMA channels which can transfer data from external I/O to internal memory vice versa and four memories DMA channels which only for internal memory usage. In performance of movement data, the DMA1/ DMA2 have a maximum transfer rate of one 32/16-bit word per two system clocks per channel and the IDMA only needs one system clock on one 32-bit word. The general signal

is one dimension (1-D); however, image is two dimensions (2-D). For this requirement, the DMA also support 2-D data movement skill. In Fig. 1, it show that the detail flow from H by V dimension to 4 by n dimension based on block type. In detail user only specify adaptable index between two blocks in vertical and horizontal direction, the DMA will move automatically block.

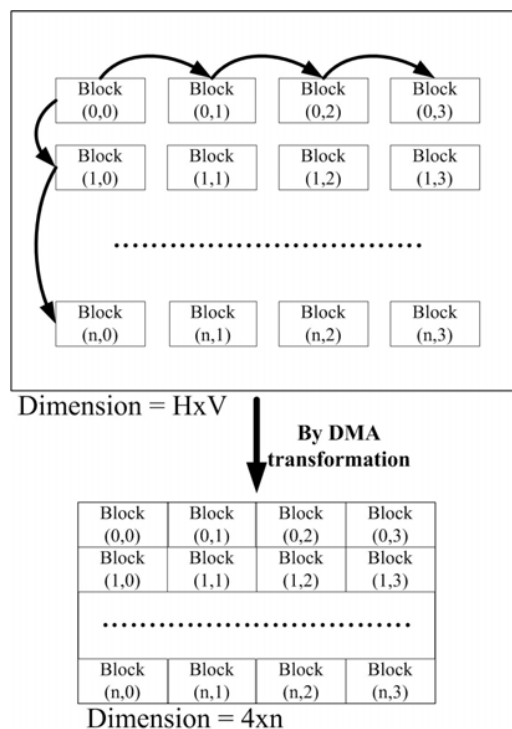


Fig. 1 The DMA scheme move 2-D data simultaneously.

The system overview can be drawn in Fig. 2, two cores fetch the same memory pool and both can communicate directly.

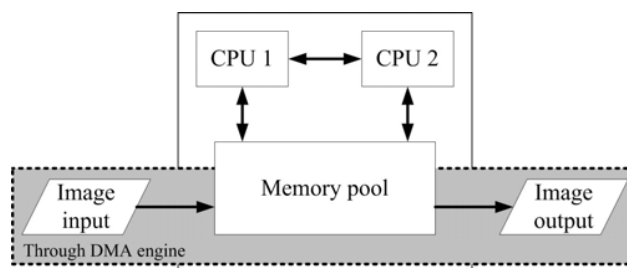


Fig. 2 The DMA engine for data input and output.

In order to improve the performance of data passing, however, ADSP-BF561 use DMA engine to implement image input and output because this architecture can efficient reduce CPU loading when image fill into memory pool and sent out. The drawback is that DMA channels will be occupied and reduce total channels. Eventually, four DMA channel and four IDMA channels can be efficient operated. Thus this system still is a suitable solution. From above description, we select Blackfin561 as target to implement our algorithm.

III. ALGORITHM FLOW

Our algorithm includes three backbones to structure a fully system which are sub-sampling, macro edge and wavelet transform. Firstly, sub-sampling provides two the same image of watermarking by 2 factor vertical direction down-sampling. Two watermarking will be separated into two domain, one is embedded on temporal domain which is only macro edge processing another is embedded on spatial domain which add macro edge processing and wavelet transform. Secondly, the macro edge is to detect most important area for embed watermarking. In general, the each edge includes un-removable information which is comfortable selection. In order to improve calculation performance, the pixel level is be replaced by macro block (MB) level. Finally, a well-known concept is that frequency domain can efficient embed watermarking and human vision can not distinguish the image has be changed. Due to wavelet transform has self-similarity property which can cooperate watermarking by sub-sampling and the lift-schemed is also proposed to fit hardware implementation, we choose this transform. All detail will be depicted as follows in detail.

A. Sub-sampling and wavelet transform

Hiding data and protection target are the main goal of watermarking. An important characteristic of image is self-similarity. The image through sub-sampling processing will be separated into two close similarity sub-image; almost all information can be extracted from any sub-image even if we lose any one sub-image. Thus sub-sampling processing can reduce possible dangerous of incompleteness image and designed attacks. We introduce here a mathematics and symbol in sub-sampling for after operation. Given a image, $X[n_1, n_2]$, $n_1 = 0, \dots, N_1-1, n_2 = 0, \dots, N_2-1$, then $X_1[m_1, m_2] = X[2n_1, 2n_2]$, $X_2[m_1, m_2] = X[2n_1+1, 2n_2]$. For $m_1 = 0, \dots, N_1/2-1, m_2 = 0, \dots, N_2/2-1$ and n_1 is vertical direction and n_2 is horizontal direction. The sub-image of $X_1[m_1, m_2]$ and $X_2[m_1, m_2]$ are obtained by sub-sampling.

For this paper, we assume that the watermarking can be expressed as $X[n_1, n_2]$ and two new sub-image, $X_1[m_1, m_2]$ and $X_2[m_1, m_2]$, is performed by sub-sampling. In the reconstruction, $X'_1[m_1, m_2]$ and $X'_2[m_1, m_2]$ are presented as reconstruction image of sub-sampling watermarking image, $X'[n_1, n_2]$ is presented reconstruction image of full watermarking.

Some meaningful data can be successfully inserted in the image frequency domain. The result and reason have been verified in the past approaches. The general methods like fast Fourier transform (FFT), DCT and DWT are used. Especially, DWT includes coexist property of temporal and spatial. Afterward, the image through DWT is separated into four sub-images which locate on LL, LH, HL and HH band in Fig. 3. Beside, the sub-image in LL band is similarity with original image. In Fig. 4, show that the image of F16 through 1-order DWT. The result is very interesting point between sub-sampling and wavelet which both can product almost the same output. The prepared watermarking through

sub-sampling feed into sub-image of LL band and original by inverse DWT, the watermarking is fully embedded on image. Thus DWT is selected on our algorithm system architecture.

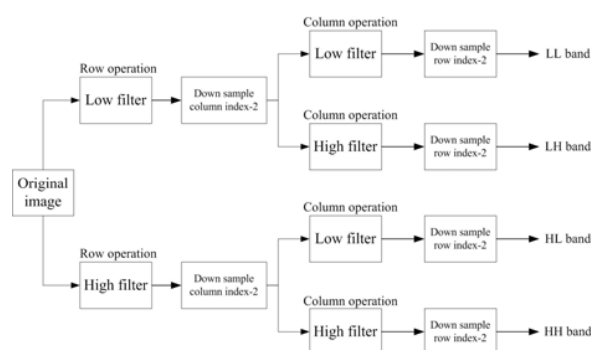


Fig. 3 The DWT scheme based on 2-D and 1-order operation

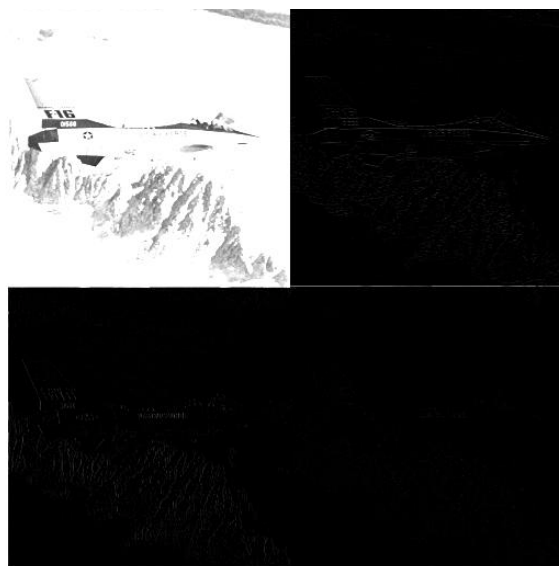


Fig. 4 The result of 1-order DWT, from left-upper to right-down by zigzag order depicts the LL, HL, LH and HH bands.

B. Macro edge and data packet

Although watermarking embed on frequency domain is invisible, the illegal section for watermarking will reduce image quality and divulge hidden information. Thus efficient area selections become an important problem which is also discussed on past approaches. In order to avoid the drawback, each algorithm will pre-search all section and calculate correlation. A good enough section for watermarking will be selected on higher correlation because this section can cover some information and keep an original image status. Generally, edge characteristic is main factor in image processing field. For example, content and object are located on edge area when the image through image processing. Median filter, Gaussian filter and Sobel operation are common method to find the edge. Although these methods can achieve a suitable result, pixel level operation is huge computation. Based on the crack and DMA property, the MB which the size is 16x16 pixels is used to replace pixel level for edge detection. The macro edge detection algorithm can be described as follows.

Step1: Divide the whole image raw data (m by n) into a two-dimensional array of 16×16 macro blocks:

$$M_{x,y} (1 \leq x \leq m, 1 \leq y \leq n)$$

If input image is color such as R, G and B three channels, each MB per channel can be expressed as follows:

$$MR_{x,y} (1 \leq x \leq m, 1 \leq y \leq n), MG_{x,y} (1 \leq x \leq m, 1 \leq y \leq n), MB_{x,y} (1 \leq x \leq m, 1 \leq y \leq n)$$

Step2: For each MB $M_{x,y}$, compute the deviation index $D_{x,y}$.

$$D_{x,y} = \frac{|M_{x,y} - M_{x,y+1}|}{|M_{x,y} + M_{x,y+1}|} \text{ for gray image}$$

$$D_{x,y} = \frac{|Mr_{x,y} - Mr_{x,y+1}|}{|Mr_{x,y} + Mr_{x,y+1}|} + \frac{|Mg_{x,y} - Mg_{x,y+1}|}{|Mg_{x,y} + Mg_{x,y+1}|} + \frac{|Mb_{x,y} - Mb_{x,y+1}|}{|Mb_{x,y} + Mb_{x,y+1}|} \text{ for color image}$$

Step3: For each MB, set the decision flags $dF_{x,y}$ by step 2.

$$dF_{x,y} = \begin{cases} 0, & D_{x,y} < \theta \\ 1, & D_{x,y} \geq \theta \end{cases}$$

where θ is the pre-defined threshold value, which is set as 0.1 in the current implementation.

Through macro edge block detection, all pixels of the MB which has 16x16 pixels when $dF_{x,y}$ equal to 1 will be embedded watermarking. The DSP has a special and powerful function which is very fast only needs one system clock to finish the work of shift-operation. In order to fit the property, how to efficient packet data is important consideration. In traditional operation of PC-based coding, the loop is a main method to packet data. However, the method will reduces system performance. Because, each index increment of loop needs one or more system cycles and only finish once in the bit replacement unless the system can provides a smart compiler and hardware architecture. Thus we separate one byte of original image data into two parts of high byte and low byte which only hold 4-bits per byte when the byte corresponds with macro edge status. The all watermarking is also separated into high byte and low byte, but it can not be calculated by macro edge detection. The high byte of macro edge will keep original image data and the low byte will insert half pixel of watermarking. The example of embedded flow is summarized in Fig. 5. Although watermarking are embedded on low byte only use half-byte replacement, macro edge property can protect all important information. The image will become footling data if edge property through attack or deleting can not be distinguished by human vision. Thus the method should keep all necessary watermarking and resist deliberate attacks.

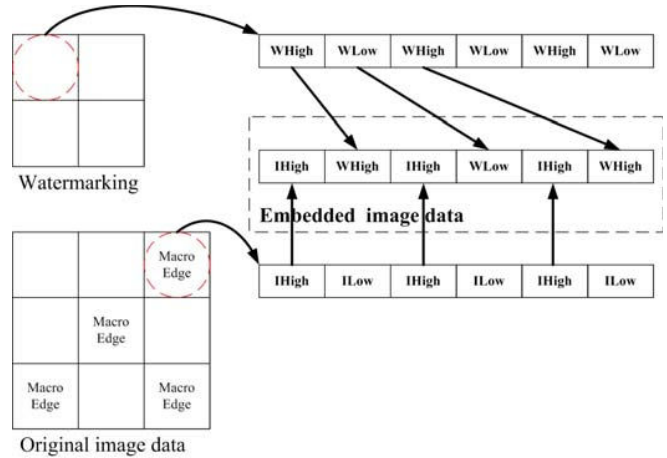


Fig. 5 An example of data packet to embed watermarking embedded.

IV. WATERMARKING INSERTION AND EXTRACTION

A full algorithm flow can be summarized in Fig. 6 and Fig. 7, which includes watermarking insertion and extraction. Beside, a balanced algorithm for dual core also is mentioned on two figures. In ADSP-BF561, we need one core and its DMA channel to deal with image input and output. Thus the core A is assigned to process the routine and to share some algorithm loading; afterward the core B will process residual works. For decoder part, only one image of embedded watermarking is used to extract watermarking and the original image is not necessary to cooperate extraction processing.

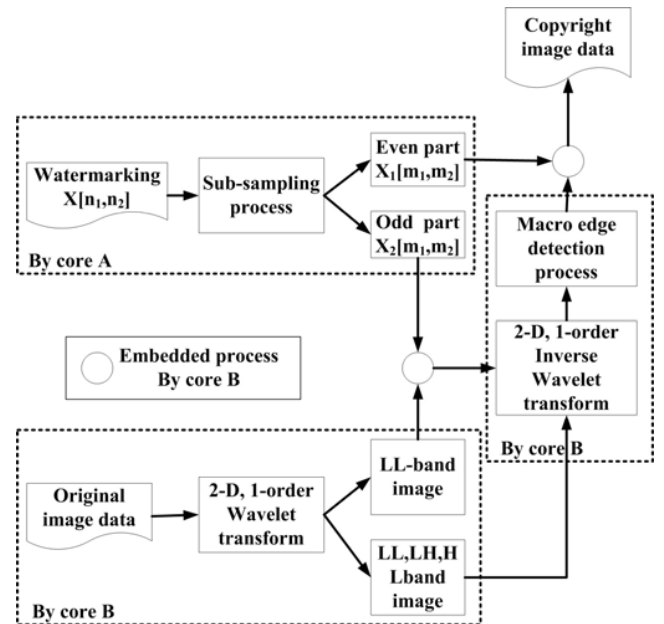


Fig. 6 The watermarking insertion flow which includes core A and core B processing respectively.

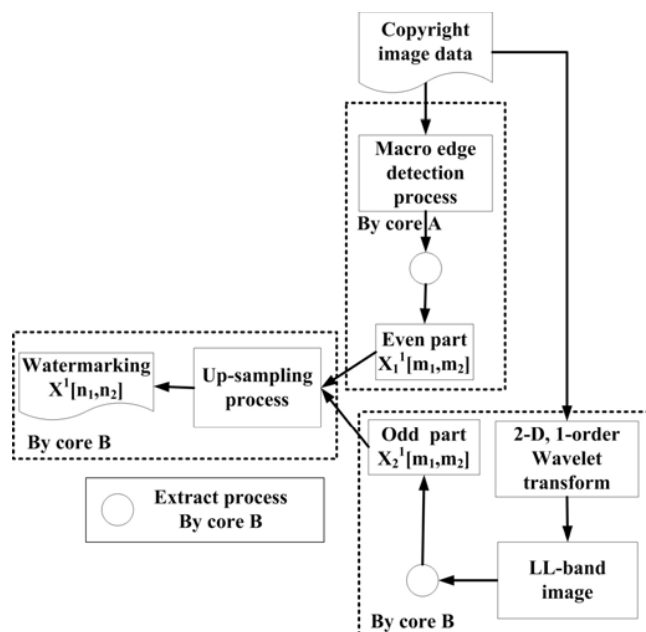


Fig. 7 The watermarking extraction flow which includes core A and core B processing respectively.

V. EXPERIMENTAL ENVIRONMENT AND RESULTS

The development environment is based on Visual DSP++ 4.0 IDE which is supported by Analog Device Company. The IDE support a lot of function which includes compiler, assembler and linker. Beside, the IDE also provides two interfaces such as JTAG and USB to connect PC and target. The execution file can be downloaded through two interfaces and directly run on target. In further, the momentary result can be transmitted to PC in immediately. Thus the status of algorithm can be analyzed by user. In the hardware specification, the maximum processing speed of ADSP-BF561 can arrive on 600M Hz for each core. The 9/7 filter is used in the wavelet transform. Although the filter is floating type and processor is fixed type, the IDE can transfer float to integer for processor operation. The standard test images which include Lena, Baboon, F16 and Pepper will be feed into our algorithm, in Fig. 8. The image size is 512 by 512 in each sample.

In the development flow, we design two projects on DSP system, one is watermarking insertion and another is extraction. Firstly, the insertion project will be opened and execute embedded watermarking algorithm and put embedded image on PC through JTAG. Then, the image will be tested on different condition such as affine transform, noise, median filter and JPEG compression by StirMark 4.0 [16]. Finally, the image will be fetched from PC and feed into DSP system by extraction project to extract watermarking.

In the StirMark, four different modes which include JPEG compression, affine transform, rotation and noise are chosen as verification items in Fig. 9. Afterward the watermarking which the size is 128 by 128 is generated by P-N generator. Through embedded process, the image quality is kept on 46 dB in average, in Fig. 10 The PSNR of watermarking which is extracted from after attack image is shown in table 2. Although

watermarking through different attack, the PSNR still has 24 dB in average. The execution time is shown in table 1 which only needs 59ms and 47ms in watermarking insertion and extraction, respectively. Thus our algorithm is very suitable for real time and embedded system application.

TABLE I THE EXECUTION TIME OF DIFFERENT STAGE IN DUAL CORE IN RESPECTIVE

| | Core A | Core B |
|----------------------|--------------|--------|
| Image Input | 8 ms | 0 ms |
| Image Output | 8 ms | 0 ms |
| Embedded Algorithm | | |
| Sub-sampling | 5 ms | 0 ms |
| Macro edge detection | 0 ms | 9 ms |
| DWT | 0 ms | 12 ms |
| IDWT | 0 ms | 12 ms |
| Data packet | 0 ms | 5 ms |
| Total time | <u>59 ms</u> | |
| Extracted Algorithm | | |
| Sub-sampling | 0 ms | 5 ms |
| Macro edge detection | 9 ms | 0 ms |
| IDWT | 0 ms | 12 ms |
| Data unpacked | 0 ms | 5 ms |
| Total time | 47 ms | |

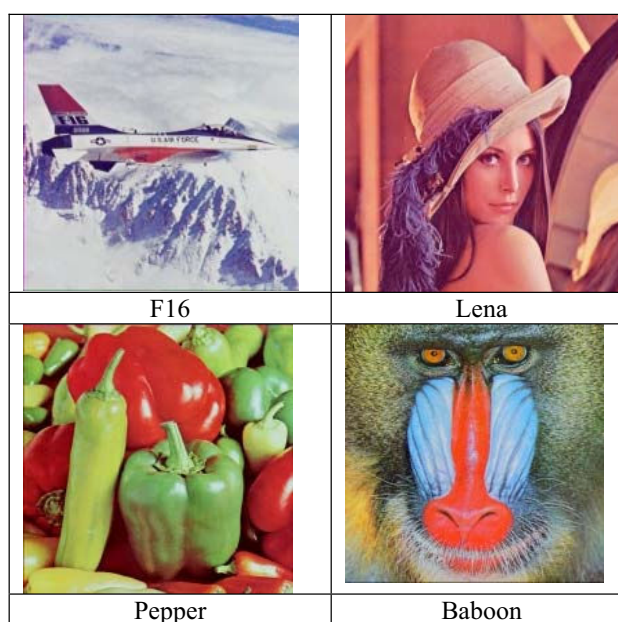
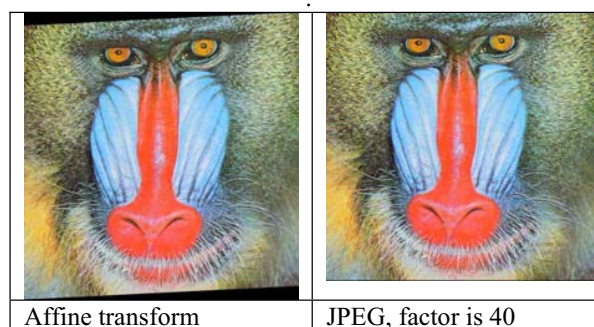


Fig. 8 Four test images for evaluation



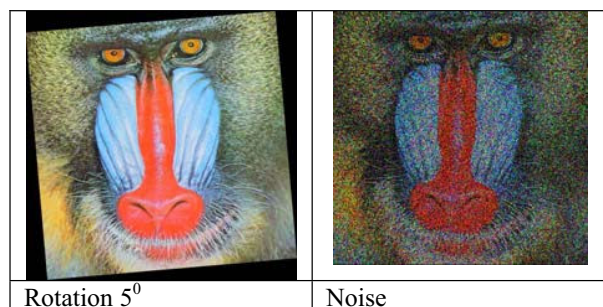


Fig. 9 Four different attacks and the baboon are used as example.

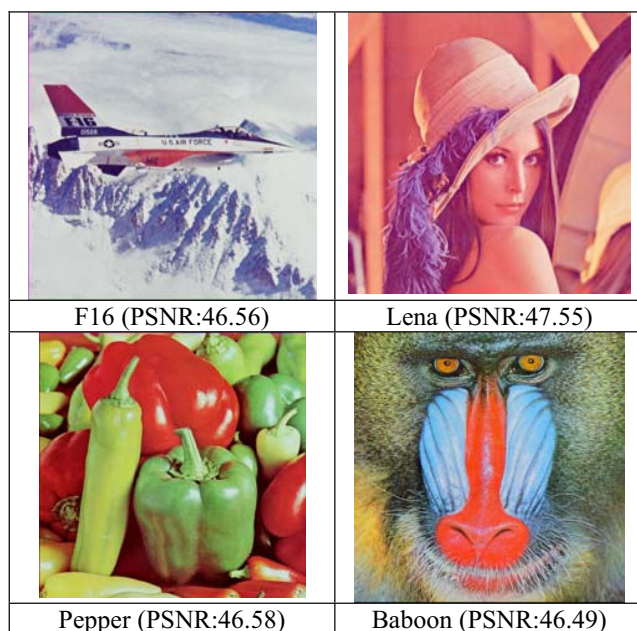


Fig. 10 Four test images after embed watermarking.

TABLE II THE PSNR OF WATERMARKING THROUGH FOUR ATTACKS WITH FOUR TEST IMAGES

| | Lena | F16 | Pepper | Baboon |
|--------------------|-------|-------|--------|--------|
| Affine transform | 24.39 | 24.33 | 24.36 | 24.49 |
| JPEG, factor is 40 | 24.64 | 24.34 | 24.34 | 24.50 |
| Rotation 5° | 24.23 | 24.30 | 24.31 | 24.49 |
| Noise | 24.53 | 24.33 | 24.26 | 24.32 |

VI. CONCLUSION

A novel image watermarking algorithm is presented together with software and hardware consideration. The algorithm is based on macro edge block and wavelet transform, the macro edge provides a suitable area for insertion and wavelet perform a frequency domain operation. It has self-similarity property from wavelet; in order to fit this advantage the sub-sampling method is chosen. On DSP consideration, the shift-operation is first entry point. Thus the data packet method is used as embedded solution. For watermarking extraction, the original image is un-useful and by macro edge process the watermarking can be extracted successful.

From the robustness test results, it can be seen the performance of the present watermarking scheme against some

of commonly used attack techniques. In general it offers reasonably good resistance against affine transform, rotation and JPEG compression, but it relatively weak under noise attack. Because the macro edge block will be changed slightly by noise addition. Although the noise performs effect, the Lena image has a visible edge and keeps a good enough quality. Beside, the insertion and extraction flow are simplest, thus it is very suitable to meet the embedded system application.

REFERENCES

- [1] F. Hartung and M. Kutter, "Multimedia watermarking techniques," in *Proc. IEEE*, vol. 87, no. July, pp. 1079-1999.
- [2] I. J. Cox, J. Kilian, T. Leighton and T. Shamon, "A secure, robust watermarking for multimedia," in *Proc. Information Hiding, First Int. Workshop*, Cambridge, U.K., pp. 185-206, 1996.
- [3] J. J. K. O'Ruanaidh and T. Pun, "Rotation, Scale and translation invariant spread spectrum digital image watermarking," *signal processing*, vol. 66, no. 3, pp. 303-317, 1998.
- [4] F. Deguillaume, S. Voloshynovskiy and T. Pun, "A method for the estimation and recovering from general affine transforms in digital watermarking applications," in *Proc. SPIE: security and watermarking of multimedia contents IV*, vol. 4675, San Jose, CA, Jan. 2002, pp. 313-322.
- [5] R. Piva, M. Barni, F. Bartolini and V. Cappellini, "DCT-based watermarking recovering without restoring to the uncorrupted original image," in *IEEE ICCP*, 1997.
- [6] R. Dugad, K. Ratakonda and N. Ahuja, "A new wavelet-based scheme for watermarking images," in *IEEE ICCP*, 1998.
- [7] W. Bender, D. Gruhi, N. Morimoto and A. Lu, "Techniques for data hiding," *IBM Sys. J.* vol. 35, pp. 313-336, 1996.
- [8] I. K. Yeo and H. J. Kim, "Generalized patchwork algorithm for image watermarking," *Multimedia Sysyem*, vol. 9, no. 3, pp. 261-265, 2003.
- [9] C. H. Lin, D. Y. Chan, H. Su and W. S. Hsieh, "Histogram-oriented watermarking algorithm: colour image watermarking scheme robust against geometric attacks and signal processing," in *Proc. IEE Vis. Image Signal Process.*, vol. 153, no. 4, Aug. 2006.
- [10] D. Coltuc and P. Bolon, "watermarking by histogram specification," in *Proc. SPIE: security and watermarking of multimedia contents II*, vol. 3657, 1999, pp. 252-263.
- [11] D. Coltuc, P. Bolon and J. M. Chassery, "Fragile and robust watermarking by histogram specification," in *Proc. SPIE: security and watermarking of multimedia contents IV*, vol. 4675, 2002, pp. 701-710.
- [12] S. Roy and E. C. Chang, "watermarking color histogram," in *Proc. Int. Conf. Image Process.*, 2004, pp. 2191-2194.
- [13] S. Lee, Y. Shu and Y. Ho, "Lossless data hiding based on histogram modification of difference images," in *Proc. Pacific-Rim Conf. Multimedia*, 2004, vol. 3, pp. 340-347.
- [14] Z. Ni, Y. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Trans. Circuits. Syst. Video Technol.*, vol. 16, no. 3, pp. 354-363, Mar. 2006.
- [15] (2007) ADSP-BF561 Blackfin Processor Hardware Reference (Rev. 1.1) [Online]. Available: http://www.analog.com/en/embedded-processing-dsp/blackfin/processor_s/manuals/resources/index.html
- [16] Available: <http://www.petitcolas.net/fabien/watermarking/stirmark/>

Yi-Pin Hsu was born in Taiwan, Taitung, in 1981. He received the B.S degrees in electrical engineering from the Private Chinese Culture University (PCCU), Taiwan, in 2003 and the M.S. degrees in Department of Mechanical Electrical in 2005 in National Taiwan Normal University (NTNU). He is now a Ph.D. candidate in electrical and control engineering at National Chiao-Tung University (NCTU), Taiwan. His research interests are in the areas of watermarking, image signal processing and DSP-based embedded system.