Implementation of Data Encryption through Dual Domain Operation on Digital Signal Processor

Yi-Pin Hsu and Shin-Yu Lin

National Chiao Tung University, Hsinchu, Taiwan

Summary
In order to protect original data, data encryption 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 digital signal processor 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.

Key words:
Watermarking, digital signal processor, embedded system

1. Introduction

Through 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 schemed 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 massage 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), 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 Taxes Instruction (TI) DM642EVM and to analyze software and hardware properties. Section III describes the watermarking embedding algorithm. An overall flow

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

combine algorithm and how to implement algorithm on DSP system are depicted in Section IV. The section V descripts a lot of balance comparison by experimental results. A conclusion is drawn in last section.

2. DSP software and hardware 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 such as TI provide different solutions. For video processing, the C6x series are main product to fit relative application; because the DSP has multi-operation units which can run on parallel execution if the data has independent characteristic.

Generally, in order to improve processing speed, most of DSP has numerous operation units. For example, TI TMSC320C64x [15] series includes two register files which has eight parallel units totally. However, the operation of addition and subtraction only are supported by four units (includes .S1, .L1, .S2, .L2). Based on two register file or dual core architecture, the data exchange becomes an essential issue. The data can be changed between file A and file B using 1X and 2X operator units. The syntax can be expressed as ".M2X" in assembly code which means data is processed in register file A by M unit and passed into some register in register file B by 2X. Thus the data can be arbitrarily exchanged between two register file and the system only delays one cycle in order to keep the transfer data safety. The redundant unit (such as M. and D.) are allocated to perform data address and some mathematical application. Although compiler can improve the program performance, assembly code is till intrinsic method. Thus using assembly code can reduce the processing time. However, TI's assembly code has special and limited rules which the program operation needs to assign a corresponding and suitable operation unit. The unit of .L1 and .L2 can operate addition instruction in respective register file, as well as the units S1 and .S2 will operate subtraction instruction. Besides, the core provides automatically 16K byte in level 1 as cache memory for data operation and 256K byte for programmable cache/RAM in level 2. Thus an efficient approach to allocate unit becomes an important work. In video processing, block based operation is expected as elementary unit. Thus the cache will be separated into multi-section and each section is one block size. In this paper, the block size is defined as 16x16 bytes (256 bytes).

Although the resources in embedded system are bounded, a real time operation system (OS) still need. Because the peripherals needs to be controlled, such as memory initialization, video device driver for capture and display image, communication port and Ethernet port. The programmer will pay a lot of attention to control system if the system con not be embedded a real time system. In order to reduce the programmer loading, TI provides a scalable real time system which calls as DSP/BIOS can allocate all necessary parameter to initial relevant devices for after application usage. Besides, a modular reference framework is designed which the framework is a set of application interface. Based on the framework, multi-thread can be created and application programs are directly implemented.

From above description, using the framework not only can fit system requirement but also construct a simple and rough program system. Our application module can be separated into three tasks which call as input task, process task and output task is depicted as following in fig. 1.

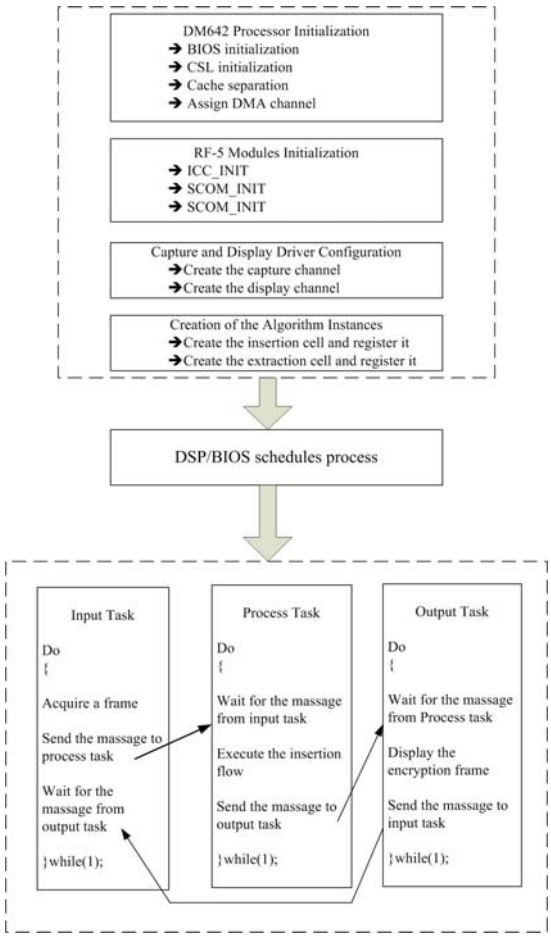


Fig. 1. The overall system flow is controlled by DSP/BIOS.

- Input task
  The input task acquires the frame from the NTSC input device, we select CMOS-based camera as target because its price is lower than CCD-based camera. Besides, the frame resolution is re-sampled to YUV420 from YUV422. Afterward the task sends the massage to the process task which describe on after explanation with the frame pointer embedded in the message. The task then waits for the massage from output task to continue.
- Process task
  The process task wait until it receives the message to insert watermarking in frame which from input device. Afterward it sends the massage to output task with the output frame pointer embedded in the massage. The task then waits for the massage from input task to continue.
- Output task
  The output task displays the frames on the NTSC output device. The acquired frame is in YUV420 and is re-sampled to YUV422. The task sends the massage to the input task to continue, and then it waits for the massage from the process task to continue.

After these initializations, the system enters the three-task system managed by DSP/BIOS scheduler. These tasks sue the SCOM modules of RF-5 to communicate with other.

3. 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 to feed 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.

3.1 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 incompletion image and designed attacks. We introduce here a mathematics and symbol in sub-sampling for after operation. Given a image, $X[n_1, n_2]$, $n_1 = 0,\ldots,N_1-1, n_2=0,\ldots,N_2-1$, then $X_1[m_1,m_2] = X[2n_1, n_2]$, $X_2[m_1,m_2] = X[2n_1+1, n_2]$. For $m_1 = 0,\ldots,N_1/-1$, $m_2 = 0,\ldots,N_2/-1$ 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. 2. Beside, the sub-image in LL band is similarity with original image.

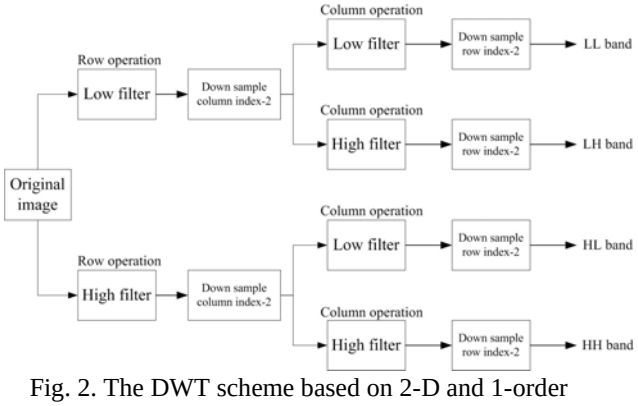


Fig. 2. The DWT scheme based on 2-D and 1-order operation

In the general image, we assume that the default dimension is 2-D and the image through DWT will perform four bands output. The result is very interesting point between sub-sampling and wavelet which both can product almost the same output, in Fig. 3. The prepared watermarking through sub-sampling feed into sub-image of LL band, the watermarking is fully embedded on image. Thus DWT is selected on our algorithm system architecture.

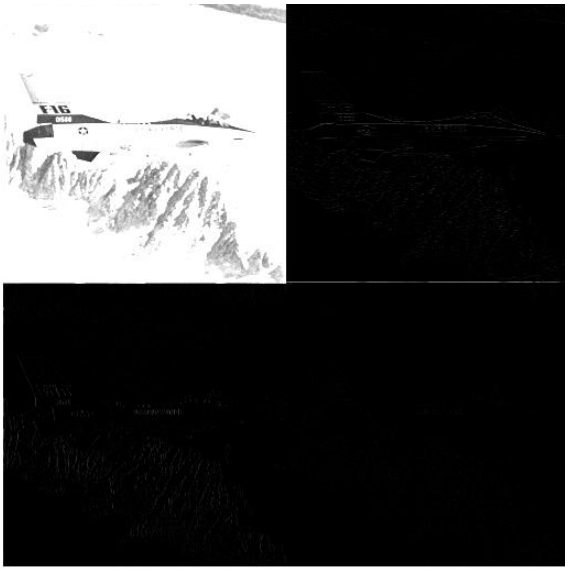


Fig. 3. The result of 1-order DWT, form left-upper to right-down by zigzag order depicts the LL, HL, LH and HH bands.

3.2 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. In H.264 video coding standard, 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 DSP loading can be shared in data movement through DMA because it move data from external/internal to internal/external directly without arithmetic calculations; 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:

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

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

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

Due to DSP architecture, the absolute operation is selected because it needs lower system cycle then square operation.

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

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

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. 4. 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.

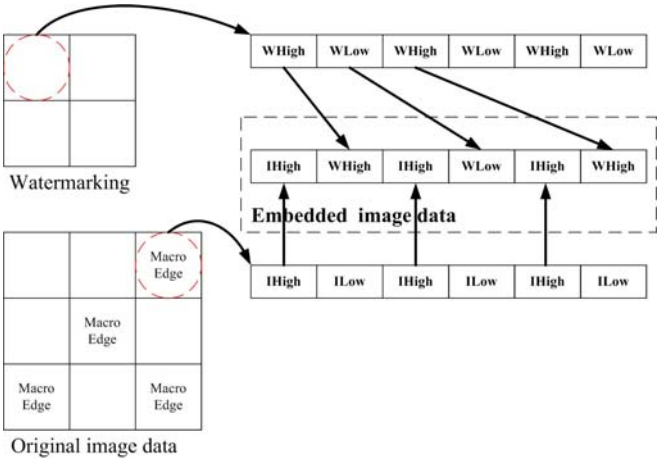


Fig. 4. An example of data packet to embed watermarking embedded.

4. Watermarking insertion and extraction

A full algorithm flow can be summarized in Fig. 5 and Fig. 6, which includes watermarking insertion and extraction. Beside, a balanced algorithm for dual register files also is mentioned on two figures.

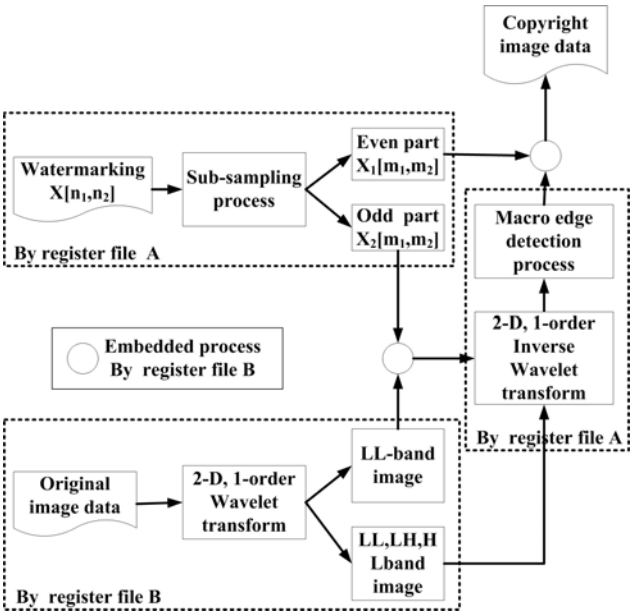


Fig. 5. The watermarking insertion flow which includes register file A and register file B processing respectively.

The register file A is assigned to process the partial routine such as sub-sampling, edge detection and IDWT; afterward the register file B will process residual works such as DWT and embedded procedures. 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.

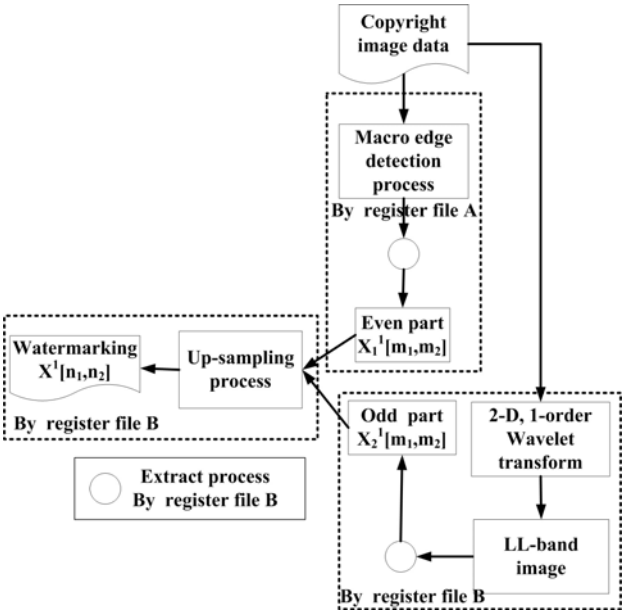


Fig. 6. The watermarking extraction flow which includes register file A and register file B processing respectively.

5. Experimental environment and results

All function such as DWT, IDWT, macro edge detection and sub-sampling are based on the property of data independent, afterward the dual register files can run on parallel The development environment is based on code composer studio (CCS), and current version is CCS v3.1 which is supported by TI. The IDE support a lot of function which includes compiler, assembler and linker. Beside, the IDE also provides interface such as JTAG to connect PC and target. The execution file can be downloaded through interface 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 DM642 can arrive on 600M Hz and 256K/32M bytes builds on the internal/external memory. In future, the 64-channels DMA is provided for acceleration

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. Besides, the TI also provides DWT and IDWT library which is based on special hardware architecture. For example, the DWT application program interface (API) can not support C5x series DSP. In future, the DWT and IDWT have been verified and tested by TI in processing performance which can archive good enough requirement. Thus the DWT and IDWT API will be used in our algorithm. The standard test images which include Lena, Boat, F16 and Pepper will be feeds into our algorithm, in Fig. 7. The image size is 512 by 512 in each sample.

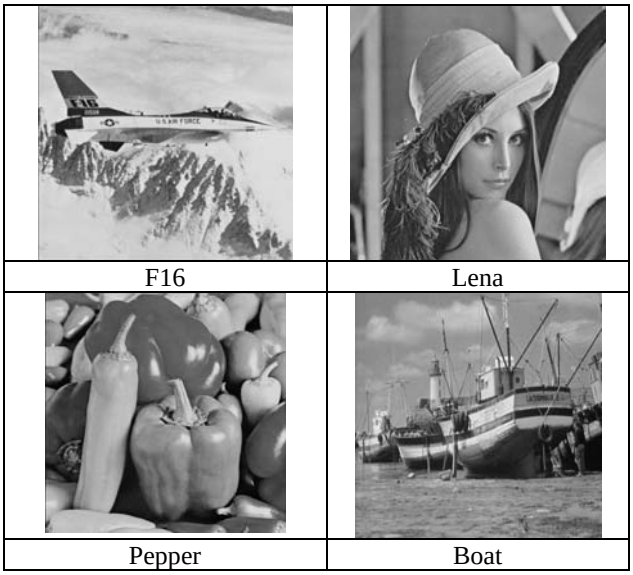


Fig. 7. Four test images for evaluation

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] which is fair and free platform. The researcher can use it to verify the robust of algorithm then reader also can refer to the valuable. Besides, the programmer can easy and fast to modify the attack item; because the platform provides all open source code. Finally, the image will be fetched from PC and feed into DSP system by extraction project to extract watermarking.

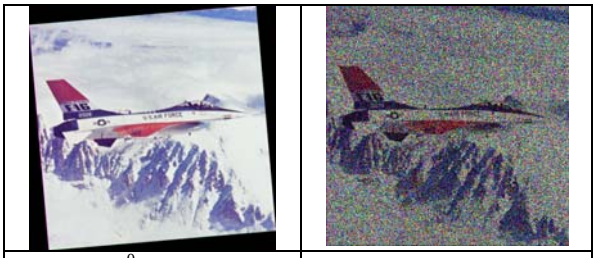
In the StirMark, four different modes which include JPEG compression, affine transform, rotation and noise are chosen as verification items in Fig. 8. Afterward the watermarking which the size is 128 by 128 is generated by P-N generator. Through embedded process, the image quality is kept on 46 dB in average, in Fig. 9, the PSNR of watermarking which is extracted from after attack image is shown in Table 1. Although watermarking through different attack, the PSNR still has 24 dB in average. The execution time is shown in Table 2 which only needs 43ms and 34ms in watermarking insertion and extraction, respectively. Thus our algorithm is very suitable for real time and embedded system application.

Table 1 The PSNR of watermarking through four attacks with four test images.

	Lena	F16	Pepper	Boat
Affine transform	24.39	24.33	24.36	24.51
JPEG, factor is 40	24.64	24.34	24.34	24.32
Rotation 5^0	24.23	24.30	24.31	24.22
Noise	24.53	24.33	24.26	24.29

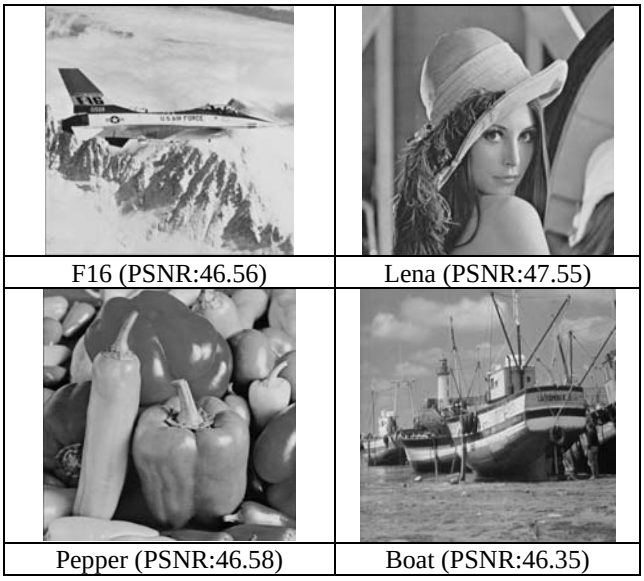
Table 2 The execution time of different stage in dual core in respective.

	Register file A	Register file B
Image System I/O		
Image Input	4 ms	4 ms
Image Output	4 ms	4 ms
Embedded Algorithm		
Sub-sampling	3 ms	0 ms
Macro edge detection	0 ms	5 ms
DWT	8 ms	0 ms
IDWT	0 ms	9 ms
Data packet	2 ms	0 ms
Total time (include image I/O)	43 ms	
Extracted Algorithm		
Sub-sampling	0 ms	3 ms
Macro edge detection	5 ms	0 ms
IDWT	0 ms	7 ms
Data unpacked	3 ms	0 ms
Total time (include image I/O)	34 ms	



Rotation 5^0	Noise

Fig. 8. Four different attacks and the F16 are used as example.



F16 (PSNR:46.56)	Lena (PSNR:47.55)
Pepper (PSNR:46.58)	Boat (PSNR:46.35)

Fig. 9. Four test images after embed watermarking

6. Conclusions

A novel image watermarking algorithm is presented together with software and hardware consideration. The algorithm is based on macro edge black and wavelet transform, the macro edge provides a suitable area for insertion and wavelet perform a frequency domain operation. On DSP consideration, the shift-operation is main entry point because it can efficient reduce system cycles. 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. Besides, the real-time OS also is used to handle the system and provides a suitable environment for application program.

Acknowledgments

The authors would like to thank the valuable support in DSP system with Prof. Chung-Yen Su at Nation Taiwan Normal University, Taipei, Taiwan.

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. Shamoon, "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] "TMS320C64/C64x+ DSP CPU and Instruction Set Reference Guide," Texas Instrument, Aug., 2006, SPRU732C.
[16] Available: http://www.petitcolas.net/fabien/watermarking/stirmark/

Yi-Pin Hsu was born in Taitung, Taiwan, 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.