Parallel Implementation of Spread Spectrum Based Oblivious Visual Watermarking Using Efficient DWT

ALI HASSAN, M. SHAMIM BAIG, Department of Cyber Security, Centre for Cyber Technology & Spectrum Management, NUST, Pakistan.

Abstract:- Watermarking is a process of adding side information in digital media for the purpose of making an assertion about the digital media or authentication of legal owner of digital media. Adding watermark in a video of high frame rate at run time requires processing of very large amount of data in very short time which is not possible on a single processor computer. A spread spectrum (SS) based watermarking technique is presented in this paper which adds watermark in the carrier digital video obliviously using efficient implementation of Discrete Wavelet Transform in parallel computing environment. Limitations of Human Visual System (HVS) are exploited to embed large amount of side information with higher strength in the carrier digital media.

Key-words:- Spread Spectrum, Oblivious, Watermarking, parallel computing environment and Human Visual System.

1 Introduction

A digital watermark is a digital pattern or signal inserted in digital multimedia data like audio, still images, video, and text documents to identify the legitimate user (authentication) or to prosecute the offender (fingerprinting)[1]. Several methods have been proposed for copyright protection [2] which can be grouped into two categories: spatial domain methods [3], [4] and transformed domain methods [5], [6]. The main break through came by publication of paper by I.J. Cox [3] in which he modeled the watermarking problem as the secure communication problem and used the Spread Spectrum (SS) technique for watermarking.

The idea presented in [3] is used in proposed scheme for adding side information into video frames by using Discrete Wavelet Transform. Limitations of HVS have been exploited to embed watermark with greater strength in carrier video frames. In this paper, proposed algorithm spreads the watermark in whole carrier frame while remaining within the visibility threshold limits of HVS thus attaining high data rates.

The main problem in adding side information to video frames, is of doing all the processing in real time. Due to high frame rate of videos and computationally extensive transformed domain watermarking techniques, these techniques can not be implemented at run time using normal desktop computers. This paper uses the idea presented in [7] and proposes an efficient way of implementing DWT using matrix multiplication on single processor computer as well as multiprocessor environment and achieved a high frame processing rate.

Section 2 gives the description of the proposed watermarking scheme, Section 3 describes the efficient implementation of DWT, Section 4 describes the parallel implementation of the proposed algorithm and results are given in Section 5 to support our claims, Section 6 concludes this paper.

2 Proposed SS based oblivious watermarking Scheme using DWT

Proposed SS based oblivious watermarking scheme has been presented in detail in [8], here we give a brief description of the scheme. Problem of watermarking multimedia applications can be manipulated by considering multimedia documents as communication channel and watermark as the message to be transmitted. Techniques like Direct Sequence Spread Spectrum (DSSS) can be used to gain substantial advantages out of them as was mentioned by I.J.Cox, [3].

In watermarking this can be interpreted as a visually imperceptible watermark, spreaded in the entire multimedia frame. Since the watermark resides in all pixels, it is likely that even when attacked at least some parts of the spectrum will always remain intact, given that usually the attacks are band limited. A watermarking scheme consists of three main blocks:

The watermark generation

The encoder (watermark embedding block)

The decoder and comparator (verification or extraction block)

2.1 Watermark Generation

Let us denote carrier multimedia frame by I, input to be watermarked by W and the watermarked frame by M. G is a generator function uses the mechanism of Spread Spectrum technique and spreads each bit of the input data over a large chip rate (cr) to add redundancy in the generated watermark and generates a randomized and amplitude adjusted [9] watermark (X) ready to be inserted into the carrier frame (I).

$$X \quad G(W,k, \) \tag{1}$$

Secret key (k) is used to seed the Pseudorandom Number (PN) generator for adding randomness in the generated watermark (X). G is shown in Fig. 1.



Fig. 1 Watermark Generation Block

2.2 Watermark Encoder/ Embedding Block

E is an encoder function. It takes edges enhanced carrier frame I, a watermark (X) and it generates a new frame, which is called watermarked frame (M), by inserting X in Discrete Wavelet transformed version of carrier frame (I). The embedding block is shown in Fig. 2.



Fig. 2 Watermark Encoder/Embedding Block

HVS is susceptible to ignore small changes in pixel values if there are large numbers of abrupt changes in pixel values in an image i-e, large number of edges. The proposed scheme exploits this idea to add watermark in edge enhanced carrier frames.

2.3 Watermark Decoder/ Recovery

A decoder function D takes the received fram e M ' (M ' is corrupted form of M) and same secret key (k) that was used in encoder and recovers

the original waterm ark W ' from the frame. In this process, an additional frame, I, can also be included which is the original and un-watermarked version of M '.M athem atically,



Fig. 3 Watermark Recovery Block

General Steps of the proposed watermarking scheme are shown in fig. 4.

```
Steps of Proposed Wavelet Domain Watermark
Embedding Process
1. Byte to Binay conversion of input data
           w[k] \{0,1\} w[s]
  BPSK
           u[k] {-1,1} w [k]
  Resampling
           b[i] = u[k] + k .cr < i (k+1).cr
4. Randomization and Amplitude Adjustment
           X = .PN.B
5. 2-D DWT of Carrier Frame
          [cA,cH,cV,cD] = DWT2(W);
6. Watermark Insertion
          m A' = cA + X; m H' = cH + X;
m V' = cV + X; m D' = cD + X;
7. IDWT of Marked Frame
           M = ID W T (m A'.m H'.m V'.m D');
Steps of Proposed Wavelet Domain Watermark
Recovery Process
1. 2-D DWT of Carrier Frame
           [rA,rH,rV,rD] = DWT2(M');
2. De-Randomization
           YA = PN.rA; YH = PN.rH;
           YV = PN.rV; YD = PN.rD;
3. De.Sampling - summation over chip rate
                      (k 1)cr 1
                                Y i], YH[i], YV[i], YD[i]
[DA,DH,DV,DD] =
                      i k.cr
4. Thresholding
           \hat{U} = sign(D)
   = A + H + V + D;
5.
6. Binary to Byte conversion
           W '[s]
                               [k] \{0,1\}
  Fig. 4 General Steps of Proposed Wavelet Domain
                Watermarking Scheme
```

3 Proposed efficient Implementation of DWT using Matrix Multiplications

In the above described wavelet based watermarking algorithm, the most computationally expensive function is DWT. In this section we describe an efficient way of implementing the DWT using matrix multiplications as was proposed by Marino [7] on a single computer as well as on a multiprocessor system. The proposed algorithm differs from [7] in the calculations of filtering matrix (W).

3.1 1-D DWT

Classically DWT is implemented using the Quadrature Mirror Theory. This scheme for calculating DWT was developed by Mallat [11]. The scheme is shown in Fig.5.



Figure 5 Mallat's A lgorithm for 1-D DWT

Here we describe the steps involved in this process. The steps for calculating forward 1-D DWT are

- 1. Filtering of input I with a lowpass filter (L) and a highpass filter (H).
- 2. Down sampling of each intermediate data by 2 to get cA and cD.

Filtering is basically convolution of the input signal with the filter coefficients and can be written mathematically as follows:

$$z[n] = \sum_{k=0}^{q-1} x[k] f[n \quad k]$$
(3)

where 0 n p+q-1 and p is length of filter and q is the length of input vector x. Equ. 3 is a vector-matrix multiplication.

In first step of DWT, input signal I is to be filtered with two filters i.e. lowpass filter and highpass filter. Both these filters can be written in matrix form, [L] and [H]. As same signal I is filtered through [L] and [H], these two matrixes can be combined to get a single matrix W as follow.

$$W = \frac{L}{H}$$
(4)

The filtering equation can be written using matrixes as

$$\mathbf{z}^{\mathsf{t}} = \mathbf{W}^* \mathbf{x}^{\mathsf{t}} \tag{5}$$

where z and x are vectors.

Second step of DWT is down sampling by two which means that every odd sample of Z will be dropped, therefore there is no point in calculating those samples. Odd rows of filter matrix (W) are dropped before calculation. The result Z will consist of two parts, first half rows will be the result of low pass filtering (cA) and the last half rows will be the result of high pass filtering (cD). We have reduced the calculation of 1-D DWT from two matrix-vector multiplications followed by two down sampling steps to single matrix-vector multiplication step, thus we have reduced the total number of computations required for the calculations of 1-D DWT.

3.2 2-D DWT

Classical implementation of 2-D DWT is basically implementation of cascaded 1-D DWT. As it has been shown that filtering and down sampling can be done in single matrix multiplication using Equ. 5, to get the final result (Y) we follow the following procedure. Writing the equ. 5 again

$$\mathbf{V}^{\mathrm{t}} = \mathbf{W}^* \mathbf{X}^{\mathrm{t}} \tag{6}$$

where X is 2-D input matrix and W is generated using equ.4. As this intermediate result is to undergo filtering and down sampling again so we put equ. 6 in equ. 5

$$\begin{aligned} \mathbf{Y}^t &= \mathbf{W}^* (\mathbf{V}^t)^t \\ \mathbf{Y}^t &= \mathbf{W}^* \mathbf{V} \\ \mathbf{Y} &= \mathbf{V}^{t*} \mathbf{W}^t \end{aligned}$$

by using equ. 6 of V

 $\mathbf{Y} = (\mathbf{W}^* \mathbf{X}^t) \ ^* \mathbf{W}^t \tag{7}$

By using equ.7 we get Y which contains cA,cH,cV and cD as follows

$$\mathbf{Y} = \begin{array}{c|c} cA^t & cV^t \\ cH^t & cD^t \end{array}$$

This efficient implementation of 2-D DWT shown in equ.7 requires three matrix multiplications to calculate the 2-D DWT as compared to the Traditional implementation which requires 6 matrix multiplications and 6 down-samplings. Although, asymptotically, the complexity of both algorithms remains same i.e. $O(n^3)$, but the total number of operations required are reduced which increases the overall efficiency of the algorithm.

3.3 Parallel Implementation of 2-D DWT

Traditional implementation of 2-D DWT of nxn m atrix using M allat's algorithm on a p processor system involves following 7-steps as shown in Fig. 6. Step 1.Downloading n/p rows to each processing element (PE). Step 2.Calculation of 1-D DWT on each n/p rows by each PE. Step 3.Uploading half calculated results to control node. Step 4.Taking transpose of the resulting matrix. Step 5.Downloading n/p rows to each PE. Step 6.Calculation of 1-D DWT on each n/p rows by each PE. Step 7.Uploading calculated results to control node.





Equ.7 is written using vector form as

$$y_k^t = \mathbf{W}^*(\mathbf{X}^* \, \boldsymbol{w}_k^t) \qquad \text{where} \qquad (8)$$

$$0 k < p+q-1.$$

Each y_k is the row of the resulting matrix Y. The equ.8 is an excellent equation for multi-processor implementation of 2-D DWT in which there will be no requirement of inter-processor communication as all the computation of y_k will be done at the same processor. This is a major advantage over the traditional implementation of DWT in multiprocessor environment which requires a large amount of interprocessor communication. Steps for calculating 2-D DWT using Efficient Implementation by using Matrix multiplications are shown in fig. 7.



Fig 7 Timing Diagram of Efficient Implementation of 2-D DWT. [G-0] Downloading X to each PE, [H-G] Calculation of first matrix multiplication. [I-H] Calculation of second matrix multiplication [J-I] Uploading results of DWT.

It can be seen in from the fig.7 and 8 that Efficient Implementation does not include extra interprocessor communication which helps in speeding up the calculation of DWT.

4 Parallel Implementation of Proposed Watermarking Scheme

The proposed scheme is implemented in parallel computing environment to meet the run time video watermarking requirements. It can be accomplished in two ways:

1. By distributing the same frame am ongst the PE 's.

2. By distributing separate frames am ongst the PE's. For a fixed frame size of a video sequence, distributing single frame am ongst PE's will have the limitation of too much to transfer and too less to compute. We can best utilize the multiprocessor environment by distributing the separate input frames to each PE and running the proposed watermarking algorithm separately on each PE. In this way, each PE will have ample amount of data to compute and we will be able to best utilize the processing power of many PEs as the results will show in the following section.

5 Results

5.1 Effect of and cr on the fidelity of recovered watermark

It has already been established that the fidelity of the recovered watermark will depend upon

the strength (alpha) as well as the redundancy (cr). Both these factors govern the quality of the recovered watermark. Fig.8 shows the results of varying and cr for proposed SS based watermarking scheme using DWT.



5.2 Visibility Threshold for strength of watermark

It was mentioned in mentioned in literature [10], that for the watermark to be invisible to a human eye, the Peak Signal to Noise Ratio (PSNR) value of the marked frame should not be less than 40dB threshold. In the proposed scheme, we have changed the formula for watermark insertion and as a consequence, PSNR visibility threshold is decreased to 27dB from 40 dB. Following graph shows the effect of varying alpha on the PSNR values of the carrier frame or cr = 32.



Fig. 9 Effect of varying alpha on PSNR of the carrier frame

Fig 10.b shows marking lena with = 8 using conventional methods proposed in [10] and Fig.10.c shows marking with = 8 using proposed method. Visible artifacts are present in Fig 11.b in the areas above the hat and other smooth areas like shoulders and in mirror.

5.3 Efficient Implementation of DWT using Matrix Multiplication

Fig.11 shows the time required to calculate the 2-D DWT of matrices of NxN dimensions on AMD Athlon 3000+ 2.0 GHz processor. The difference between performances of the two methods is quite clear for matrices of large dimensions. Fig.11 shows that the Efficient Implementation of 2-D DWT using Matrix multiplications is computationally less

expensive as compared to the Traditional Implementation.



Fig. 11 Timing Comparison b/w Classical 2-D DWT and Efficient 2-D DWT Implementation

5.4 Parallel Implementation of Proposed Watermarking Scheme

Timings for parallel implementation of proposed watermarking scheme by distributing the single frame am ongst the PE 's are show n in Fig. 12.



Fig.12 Timings for Embedding and Recovery Processes using proposed Scheme

Best timing results are obtained when we use 16 PE simultaneously with which we can process almost 11 frames per second. Thus, one can deduce too much data to transfer and too less data to compute.

Therefore, instead of distributing the algorithm, if we distribute the video frame amongst each PE. Theoratically, by using 64 processors, we should be able to process 64 frames per second. Experimentally we have achieved processing of 50-53 frames per second using 64 PE. This is a very high processing rate, which can enable the use of watermarking schemes on live videos to transfer side information at run time.



b. Marked Image using conventional Method

R

c. Marked Image using Proposed Method

Fig. 10 Marking Lena Image With = 8

D DWT proposed in this paper have better performance than the traditional implementation of 2-D DWT using M allat's A lgorithm .. The parallel implementation of the proposed scheme is efficient enough to be applied on run time videos at high frame rates.

References:

6. Conclusions

[1] D avid A ucsm ith, "A ttacks on C opyright M arking System s," Second workshop on information hiding, in vol. 1525, pp. 218-238, 14-17 April, 1998, Portland, Oregon, USA.

It can be concluded from the results that the

proposed SS based watermarking scheme is a

workable scheme and efficient implementation of 2-

[2] G. Langelaar, I. Satyaw an and R. Lagendijk, "W aterm arking digital im ages and video data" *IEEE signasl processing magazine*, vol.17,no.5,pp.20-46, spe 2000.

[3] I.J. Cox, J. K ilian, T. leighton and T. Shamoon, "Secure spread spectrum watermarking for multimedia", *IEEE Transaction on Image Processing* 8, pp.58-68, 1999.

[4] S.Maity, M.Kundu, "Robust and Blind Spatial Watermarking in Digital Image ", *Proc. 3rd Indian Conf. on Computer Vision*, *Graphics and Image Processing (ICVGIP '2002)*, pp. 388 -393 , Ahmedabad, India, 16-18th December, 2002.

[5] Piva A . "DCT-based watermark recovering without-resolving to the uncorrupted original im ages" *in Proc of ICIP*'97,1997:520-523

[6] A lturki Faisal, M ersereau Russell,"Secure, blind im age w aterm arking technique for copyright protection", *in the proceedings EUSIPCO*, Volume III pp 101-104, 2002.

[7] F. Marino, *et al.* "A Parallel Implementation of the 2-D Discrete Wavelet Transform without Interprocessor Communications," *IEEE Transactions On Signal Processing*, VOL. 47, NO. 11, November 1999.

[8] A . H assan, S . B aig, "Spread Spectrum based 0 blivious V isual W aterm arking",

[9] M. Kutter, F. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," *Proc. of SPIEE197*, Feb. 1997, pp. 518-526.

[10]S.Maity, M.Kundu, "Robust and Blind Spatial Watermarking in Digital Image ", *Proc. 3rd Indian Conf. on Computer Vision*, *Graphics and Image Processing (ICVGIP '2002)*, pp. 388 -393 , Ahmedabad, India, 16-18th December, 2002.

[11] S. M allat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Pattern Anal. and Machine Intell.*, vol. 11, no. 7, pp. 674–693, 1989.



a. Carrier Image