# Project Report for ELEC 5330 -Video Signal Processing

*Name*: Sunil P. Jaiswal *Student ID*: 20083668 *Program*: PhD in ECE

*Course Instructor*: Prof. Oscar C. Au

$24^{th}$ December'2012

The Report is based on a paper titled "Reversible Image Watermarking Using Interpolation Technique" ($RIWIT$). This work was published in "IEEE Transactions on Information Forensics and Security" in March 2010.

Reversible invisible watermarking not only provides the protection of the copyright by embedding the assigned watermarked into the original image but also can recover the original image. Digital Watermarking defines technologies that hide information in form of numbers, text, image etc in digital media, such as image, video or audio. The increasing use of Internet as an open medium to share work has given a hike in complaints against piracy. Hence adding watermarks in digital data helps in preserving originality of the data and resolving the copyright issues .Reversible image watermarking in particular is of great importance in the fields where we need to restore the cover image precisely such as in military, medical science etc.

Reversible invisible watermarking is a special kind of digital watermarking. Reversible invisible watermarking ($RIW$) have been classified into three categories :

1. Reversible watermarking using data compression,
2. Reversible watermarking using difference expansion ($DE$),
3. Reversible watermarking using histogram operation.

In this report, we will focus on watermarking scheme using difference expansion ($DE$). In general, $RIW$ using difference expansion, can be divided into two phase. In first phase, input image (cover image) is predicted to generate a residual (error) image and in second phase, bits (watermarked) are embedded in the residual image by using $DE$ method. The watermarked image is then sent to the decoder. And recover the original image (cover image) after extracting the embedded bits. The embedding capacity (second phase) is highly dependent on the prediction accuracy of the predictor used in first phase. Thus several researchers work on increasing the prediction accuracy of the predictor which is an important issue of reversible watermarking scheme using $DE$ method.

## 1 Contribution of RIWIT

The paper titled "Reversible Image Watermarking Using Interpolation Technique" ($RIWIT$) used existing difference expansion ($DE$) method for embedding the watermarked data. In $RIWIT$, the authors have proposed a method for reversible image watermarking that uses interpolation technique for prediction. Interpolation technique gives an advantage of using Symmetrical Predictor Structure in comparison to the Non-Symmetrical Predictor Structure (using only causal pixels) used by the conventional methods for predicting input image that ensures high embedding capacity.

The aim of this paper ($RIWIT$) is to increase the embedding capacity while retaining the quality of the watermarked image. This is achieved as follows:

1. By exploiting the correlation of the pixels to the maximum extent to get low interpolation error values in first phase.
2. Interpolation method follows symmetrical structure and can utilize all the original pixels (both causal and non-causal) for prediction.
3. Thus, in second phase, it can achieve high embedding capacity by using existing $DE$ method.

# 2 RIWIT Algorithm

The working procedure of $RIWIT$ at the encoder side can be divided into two phase. It can be explained as follows:

## 2.1 Prediction Stage - First Phase

Before prediction, $RIWIT$ undergoes a preprocessing stage to avoid underflow/overflow.

### 2.1.1 Under/Over flow control - Preprocessing stage

Let $I$ be an original 8-bit image. Embedding of data bit (Second phase) is done by shifting prediction error may results under/over flow. To avoid it, $RIWIT$ modify the values of extreme pixels of $I$ by (1). Thus the new modified image ($I_m$) has the pixel range from [$Q$,$255 - Q$], where $Q$ is a constant which depends on the degree of embedding. The modification is recorded in overhead information and send it to decoder to ensure reversibility as done in method described in [1].

$$I_m(i,j) = \begin{cases} Q & \text{if } I(i.j) \leq Q \\ I(i,j) & \text{if } Q < I(i,j) < 255 - Q \\ 255 - Q & \text{otherwise} \end{cases} \tag{1}$$

### 2.1.2 Generating Interpolated residual Image

Initially down-sample the cover image ($I_m$) of dimension $W \times H$ by a factor of two; row wise and column wise. Due to direct down sampling operation, a low resolution version LR of the dimension $W/2 \times H/2$ is obtained. The low resolution image (LR) is then interpolated to High Resolution Image (P) using a Two Pass algorithm [2].

1. The odd-odd positioned pixels denoted by black dots in Fig.1 are copied from LR image as shown in (2) and these pixels are used for predicting rest of the pixels.

$$P(2i-1, 2j-1) = LR(i,j) \tag{2}$$

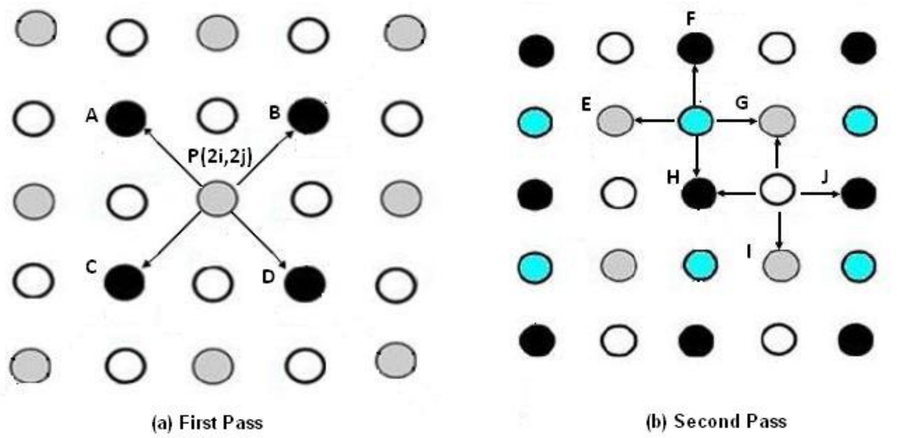


Figure 1: Two Pass Interpolation Algorithm. Black dots represent original pixel of LR image. Rest of the pixel are interpolated by [2]

   (a) In first pass, P(2i,2j) or the even-even pixels denoted by gray dots in Fig.1(a) are predicted by its neighboring pixels. Initially, its neighboring pixels are divided into 2 orthogonal sets (at $45^0$and $135^0$).Each set compute $P_{45}$ and $P_{135}$.

$$P_{45}(2i, 2j) = (B + C)/2 \tag{3}$$

$$P_{135}(2i, 2j) = (A + D)/2 \tag{4}$$

(b) The final value of the predicted pixel P(2i,,2j) is considered to be the weighted sum of $P_{45}$ and $P_{135}$.

$$P(2i, 2j) = w_{45} P_{45}(2i, 2j) + w_{135} P_{135}(2i, 2j) \quad (5)$$

where $w_{45}$ and $w_{135}$ are the prediction coefficient of $P_{45}$ and $P_{135}$ receptively.

(c) The weights are computed as :

$$w_{45} = \frac{Var(e_{135})}{Var(e_{45}) + Var(e_{135})}; w_{135} = 1 - w_{45} \quad (6)$$

Where $e_{135} = P_{135} - I$ and $e_{45} = P_{45} - I$

(d) For computing Variance (Var), we consider m as the mean value given by:

$$m = (P_{45} + P_{135})/2 \quad (7)$$

$$Var(e_{45}) = \frac{(P_{45} - m)^2 + (B - m)^2 + (C - m)^2}{3} \quad (8)$$

$$Var(e_{135}) = \frac{(P_{135} - m)^2 + (A - m)^2 + (D - m)^2}{3} \quad (9)$$

2. In second pass P(2i,2j-1) and P(2i-1,2j) (denoted by blue and white dots resp) in Fig. 1(b) are predicted in the similar manner using the two orthogonal sets at $0^0$ and $90^0$.

## 2.2 Difference Expansion based watermarking - Second phase

Interpolated error image (E) can be obtained by subtracting the interpolated image from original image ($I_m$) shown in (10).

$$E = I_m - P \quad (10)$$

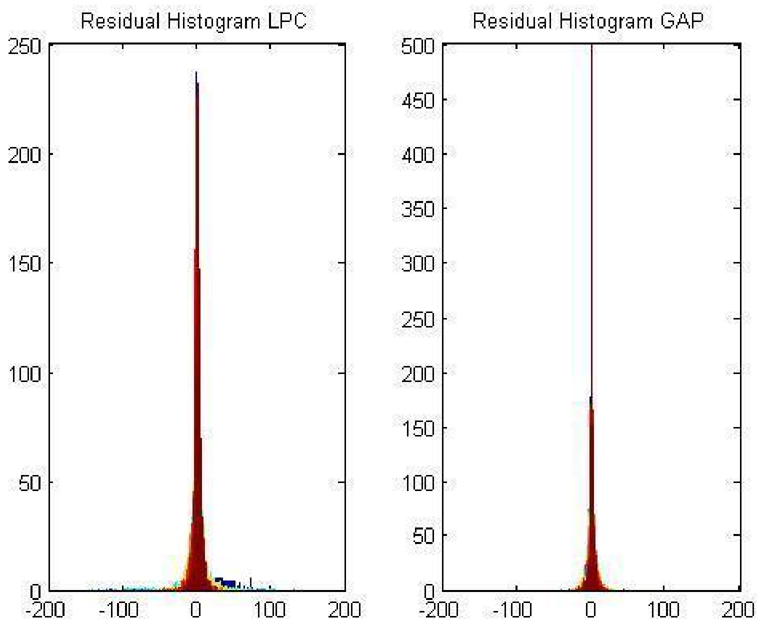


Figure 2: Residual Histogram Analysis of Linear Predictor

The embedding scheme is based on shifting of interpolation errors [2]. The watermarking procedure of $(3/4)^{th}$ interpolation error samples ($E$) are as follows: Let $I_m$ be original Image and $I_w$ be output watermarked image. The given data stream (Binary bits) or watermarked stream is embedded into error Image ($E$) depending on it's error sample value. And this is done by modifying the interpolation error sample as shown in (11).

$$E'(i,j) = \begin{cases} E(i,j) + b \times sign \times Q & -Q \leq E(i,j) < \text{Q} \\ E(i,j) + b \times Q & \text{if } E(i,j) = 0 \\ E(i,j) + sign \times Q & \text{otherwise} \end{cases} \tag{11}$$

Here $b$ is to-be-embedded bit (0 or 1), $Q$ represents degree or order of embedding and sign represents $sign(E(i,j))$ ($+ve$ or $-ve$). The watermarked output is given by (12). It can be clearly seen that the embedding capacity is totally dependent on t

$$I^w(i,j) = P(i,j) + E'(i,j) \tag{12}$$

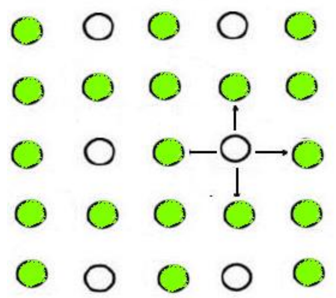

Figure 3: Green dots and white dots are watermarked sample and non-watermarked samples respectively

#### 2.2.1 Watermarking Process of remaining $(1/4)^{th}$ Pixel

However, by the above two phase, only $(3/4)^{th}$ pixels of $I_m$ gone undergoes watermarking process. The watermarking process for remaining original $(1/4)^{th}$ pixel (non-watermarked) can be described as follows:

1. Since by (11) and (12), each pixel of $I_m$ is modified at most by 1 through interpolation-error expansion, thus high image quality is preserved.

2. The non watermarked samples ($(1/4)^{th}$ pixel of $I_m$) can be interpolated using watermarked samples along two orthogonal directions as shown in Fig. 2. and the corresponding interpolation-error can be easily obtained.

3. Then, remaining $(1/4)^{th}$ interpolation error undergoes watermarking process shown in (11).

Thus. the whole image is scanned and final watermarked image ($I_w$) is obtained. Then it is send it to decoder.

## 3 De-embedding or Extraction Process at Decoder side

In this section, steps for extraction of hidden watermark and lossless recovery of original image are detailed. Initially, $(1/4)^{th}$ pixel of $I_w$ undergoes de-embedding process.

1. Interpolation and Error estimation of $(1/4)^{th}$ pixels of $I_m$:

   At decoder, $(1/4)^{th}$ pixels of $I_m$ are interpolated using watermarked samples along two orthogonal directions as shown in Fig.2. and the corresponding interpolation-error can be easily obtained. Let $P(i,j)$ is the interpolated value at spatial location $(i,j)$. Thus watermarked error at $(i,j)$ can easily be obtained by (13).

$$E^w(i,j) = I^w(i,j) - P(i,j) \tag{13}$$

2. De-embedding process of $(1/4)^{th}$ pixels of $E^w$: The hidden bit and original pixel ($I_m$) value are computed as below

$$E(i,j) = \begin{cases} E^w(i,j) \text{ and b} = 0 \\ \quad \text{if } -Q \leq E^w(i.j) < \text{Q} \\ \\ E^w(i,j) - sign(E^w(i,j)) \times Q \text{ and b=1} \\ \quad \text{if } -2 \times Q \leq E^w(i,j) < -Q \\ \quad \text{or } Q \leq E^w(i,j) < 2 \times Q \\ \\ E^w(i,j) - sign(Ew(i,j)) \times Q \\ \quad \text{otherwise} \end{cases}$$

Thus data stream (Embedded data) is extracted and original pixel value is obtained by (14).

$$I_m(i,j) = E(i,j) + P(i,j) \tag{14}$$

3. Under/over flow recovery:- We can get $I_m(i,j)$ by (14) and the original value ($I_m$) is extracted using the under/over flow information [9].

Once we get $(1/4)^{th}$ original samples ($I_m$), then our task is to get back the remaining embedded data and $(3/4)^{th}$ original samples of $I_m$. This is explained as follows:

1. Interpolate the remaining $(3/4)^{th}$ samples with the help of $(1/4)^{th}$ original samples ($I_m$) as it was done at encoder.

2. Let $P(i,j)$ be the interpolated pixel at $(i,j)$. Then follow the above steps, and extract the embedded information and thus recover the original samples ($I_m$) easily.

Thus all the embedded information or watermarked value are extracted and recover the original image. Thus the interpolation based watermarking algorithm ($RIWIT$) is reversible.

# 4 Overview of Proposed Algorithm

The aim of proposed algorithm is to further reduce the prediction errors so as to achieve higher payload capacity. In order to achieve this, we improve the correlation of the interpolated image with the original image by editing the Prediction phase of $RIWIT$. Proposed Algorithm can be divided into two phase.

## 4.1 Prediction and Watermarking of $(3/4)^{th}$ pixels of original Image - First Phase

We First compute the Interpolation Error Image (E) by subtracting interpolated image( P) from original image (I) as it is done in $RIWIT$ algorithm. $E = I - P$.

Now we improve the prediction accuracy of the Interpolation algorithm by reducing the interpolation errors (E). The modified/corrected Image ($I_C$) is calculated by computing the average interpolation error ($E_{avg}$) using the neighboring causal pixels of E(i,j). Causal pixels are used to ensure reversibility at decoder.

$$E_{avg} = (E(i-1,j-1) + E(i-1,j) + E(i-1,j+1) + E(i,j-1))/4 \tag{15}$$

$E_{avg}$ is the average of the errors that given interpolation algorithm has caused to the neighboring pixels (P(i,j)). In other words, performance of the interpolator is biased towards the pixels (P(i,j) by the amount of $E_{avg}$. Hence removing the biased value from each of the interpolated value from P will improves the performance and thus will decrease the interpolation error. Thus the new modified interpolated image is given by (16).

$$I_C(i,j) = a \times E_{avg} + P(i,j) \tag{16}$$

Here $a = 0.4$ is kept and is based on experimental basis. Thus after improving the interpolated image, a similar procedure of watermarking can be done as it is done in $RIWIT$. And remaining part ($(1/4)^{th}$ pixels of P) can undergo a similar process as it is done in $RIWIT$.

### 4.2 Simulation Result

We have implemented proposed algorithm and has shown a subjective quality of proposed algorithm and $RIWIT$ in Fig. 4 and Fig. 5 respectively. The proposed algorithm has 0.8 dB higher PSNR than that of $RIWIT$, but it's not the end of world. Still a lot of improvement can be done to make a better subjective quality as well as increasing the payload capacity.


Figure 4: Proposed Algorithm


Figure 5: By RIWIT

# 5 My Comments/Views on $RIWIT$/Conclusion

Since, embedding capacity is dependent on prediction algorithm. Thus it is very essential to maintain a better predictor. $RIWIT$ method uses an interpolation technique to generate interpolation-errors, to embed data by additive expansion. The author had used the symmetrical predictor structure for prediction/interpolation. But still major improvement can be done to improve prediction accuracy.

1. This algorithm had used symmetrical predictor structure which consist of 4 pixels to predict/interpolate the image. Whereas, we can use use symmetrical predictor structure consists of 8 neighboring pixels. Thus extracting more information (nature of pixel - edges/smooth) for prediction and leads to low prediction error.

2. The algorithm works either for edges at $45^0$ and $135^0$ or for edges at $0^0$ and $90^0$. However, we can propose an algorithm which can works for edges of any direction.

3. The down-sampling method used in $RIWIT$ is not dependent on the interpolation algorithm. Thus instead of direct down-sampling, we can propose a down-sampling method which is content adaptive and is dependent on interpolation algorithm. And the same can be done at the decoder to ensure reversibility.

4. $RIWIT$ method uses two pass interpolation algorithm. Thus it propagates error from first pass to second pass. To reduce error propagation, a one pass interpolation algorithm can be proposed.

# References


[1] Conotter, V.; Boato, G.; Carli, M.; Egiazarian, K., , "High capacity reversible data hiding based on histogram shifting and non-local means,". in *Local and Non-Local Approximation in Image Processing, 2009. LNLA.*

[2] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," in *IEEE Trans. on Image Processing, vol. 15, no. 8, pp. 2226-2238, Aug. 2006.*

[3] J. Tian, , "Reversible data embedding using a difference expansion, in *IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890896,2003.*

[4] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *IEEE Trans. on Circuit and System for Video Technology, vol. 16, pp. 354-362, 2006.*

[5] L. Luo, Z. Chen, M. Chenm, X. Zeng, X. Zhang, "Reversible Image watermarking using interpolation tech- nique," in *IEEE Trans. Inf. Forensic Security, vol. 5, no. 1,pp. 187-193,march 2010..*