IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 16, 2018, accepted January 24, 2018, date of publication February 8, 2018, date of current version March 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2803301

Hybrid Predictor Based Four-Phase Adaptive Reversible Watermarking

MUHAMMAD ISHTIAQ[1], WAQAR ALI[1], WASEEM SHAHZAD[1], MUHAMMAD ARFAN JAFFAR[2], AND YUNYOUNG NAM[3]

[1]National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan
[2]College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia
[3]Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Yunyoung Nam (ynam@sch.ac.kr)

This work was supported by the Soonchunhyang University Research Fund and also supported by the "ICT Convergence Smart Rehabilitation Industrial Education Program" through the Ministry of Trade, Industry & Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT).

ABSTRACT Reversible watermarking has gained importance due to increased involvement of digital media in sensitive fields, such as medical and law enforcement. We propose a prediction error expansion-based watermarking scheme that allows embedding reversible watermark in the image with low distortion. Research work proposes four-phase representation of image which allows exploitation of larger prediction context. We have also proposed a hybrid predictor that helps enhance the prediction accuracy. To reduce image distortion at lower capacity payloads, we use sorting of estimated prediction errors through sorting of prediction context variances. For improvement at higher capacity payloads, adaptive embedding is used to determine whether to embed single or two bits in a given prediction error. The results are compared against some state-of-the-art techniques in the field and show promising results.

INDEX TERMS Adaptive watermarking, error-expansion, prediction, reversible watermarking.

I. INTRODUCTION

Image watermarking refers to the process of embedding some data into an image for its authentication or protection. Reversible watermarking is the kind of watermarking where the watermark as well as original data can be completely restored without even a single bit error. Though watermarking has been around for centuries first idea for *reversible* watermarking emerged in 1997 [1]. In reversible watermarking an authorized person can easily remove the watermark to reverse the watermarked image back to its original version. Different methods have appeared in the literature for reversible watermarking. *Difference Expansion* is one of the most efficient methods for reversible watermarking in terms of low distortion and high payload capacity. It makes use of correlation between adjacent pixels. The difference between the two adjacent pixels is shifted to add the data bit to it. For example, Tian used difference between two pixels and the mean between them to embed the information. All the pixels in the image cannot be watermarked as some of them might overflow or underflow. To recognize the pixels which contain the embedded values in them, the difference expansion relies on location maps. Main focus of research under difference expansion field has been on minimizing the size of this location map.

Prediction Error Expansion emerged from difference expansion. Prediction error expansion uses a predictor to use neighborhood of a pixel to predict its value. The predicted value is then subtracted from the actual value to get a prediction error. That prediction error value is expanded to embed the data bit. The main difference of prediction error expansion with difference expansion is that prediction error expansion uses a larger prediction context rather than using difference of only two adjacent pixels. Having a larger prediction context allows better correlation between prediction context and the pixel. As a result of better correlation the prediction error in the predicted value is lowered as compared to difference expansion. Expansion of lower prediction errors generates lesser distortion in the image [2]. So the main focus of prediction error expansion research is on selection of good predictors and prediction contexts. Similar to its predecessor difference expansion, location maps are also required in prediction error expansion. Thus map representation are also thus considered for optimization in many of the research works in literature.

Watermarking requirements for medical, legal, military, trust chain based businesses and other sensitive fields are slightly different from that of an average user. These are fields where the data precision is as important as the authenticity

of being from the correct source. Traditional image watermarking ensures that the image is from the correct source and is not modified ever since being watermarked. But the watermarking initially makes changes to the image itself for embedding the watermark in to the image. This calls for reversible watermarking methodology so that the image can be recovered with hundred percent accuracy and can be exchanged with an embedded watermark.

A reversible watermarking method is the one which is able to extract the embedded message and completely retrieve the original image from the watermarked image. However there are also some desirable features such as low image distortion and high payload capacity. *Image distortion* is the modification in the image caused by embedding the watermark. Image distortion is measured in terms of peak signal to noise ratio of the watermarked image with original image. *Payload capacity* is the size of payload that can be embedded. Payload capacity is measured in terms of bits per pixel (bpp) of image. However, it is important to remember that image distortion and payload capacity are inversely proportional.

In this study we propose a four phase prediction error expansion based invisible and fragile watermarking scheme. This watermarking scheme provides a larger prediction context of 3×3 window for every embedding candidate pixel thus is more capable of using more sophisticated predictors thus reducing the prediction errors. We also tailor some well known techniques from state of the art watermarking schemes from both difference expansion and prediction error expansion to reduce the image distortion. Results show reduction in image distortion as compared to the current state of the art techniques (specially for smaller payloads).

In section II, we discuss the literature review with special emphasize on prediction error expansion and its predecessor difference error expansion. Section III. describes our proposed method including encoding and decoding schemes. It also describes the approaches used to reduce image distortion and four phase embedding and the benefits of these approaches. Section IV discusses the experimental results and compares them with the existing state-of-the-art techniques in literature. We then present a short discussion of our work. Conclusions are drawn in section V

II. RELATED WORK

Though non-reversible watermarking has been known to mankind for centuries. The earliest registered idea of *reversible* watermarking is traced back to the Barton patent in 1997 [3]. Barton's method was based on simple lossless image compression to create space for embedding data. Though the idea was revolutionary but was not very capable in terms of low image distortion and high payload capacity. Difference expansion and prediction error expansion emerged in early 21^{st} century. These methods are more capable at lower image distortion and higher payload capacity. Difference expansion emerged in 2003 and became popular because it provided higher capacity and lower distortion than lossless compression based reversible watermarking schemes prevalent at that time. A year later a very similar but more flexible approach namely prediction error expansion was introduced [4].

Image distortion caused by a watermarking scheme is defined as the modification in the image caused due to watermarking. Image distortion is measured in terms of peak signal to noise ratio (PSNR) of watermarked image with original image.

Payload capacity of a watermarking scheme is defined as size of payload that can be embedded in an image of a given size. Payload capacity is measured in terms of bits of payload per pixel of image (bpp).

Difference expansion is a reversible watermarking scheme that embeds payload data in differences of consecutive pixels. First the difference of consecutive pixels of the image are calculate for the complete image. Next the differences are expanded to embed the data bits of payload.

Prediction error expansion is a reversible watermarking scheme that embeds payload data in the difference of a pixel's actual value and its predicted value. First prediction is made for pixels of the image, then the difference of the predicted value and the actual value is calculated. This difference between predicted value and actual value is called prediction error. Prediction error is expanded to embed the data bit. Prediction error expansion schemes are more flexible descendant of difference error expansion watermarking schemes.

Peng *et al.* [5] have proposed block level adaptive embedding aiming mainly to achieve lower distortions particularly at higher capacity levels (while compromising at some level on increase in distortion for lower capacity levels). Adaptive embedding is done using an equation to determine the capacity parameter k by defining it as inversely proportional to that of variance of the pixels in an $n \times n$ pixels block.

The method first divides the image into non-overlapping blocks, then calculates the capacity parameter k for all the blocks. Next the location map is collected, optionally encoded with Huffman encoding scheme (if encoding gives favourable compression) and then compressed with another lossless compression scheme (name not mentioned in the paper). The location map and watermark data are then embedded into the blocks to generate the watermarked image. Adaptive approach suggested by the authors gives much better results at higher capacity embedding, whereas displays slightly worse performance at capacities lower than 1 bpp.

Xuan *et al.* [6] have proposed use of double threshold on histogram of prediction errors. First threshold called embedding threshold is used to select only those pixels for watermarking whose prediction error does not exceed the predetermined threshold. Second threshold called fluctuation threshold is to select only those pixels whose neighbor fluctuation does not exceed the said threshold. Also image gray level histogram modification is also used to concentrate the prediction error histogram towards the middle to avoid underflow and overflow problems. They have used eight pixel neighborhood for prediction using the sliding

3 × 3 window. For prediction they have used a weighted mean prediction that is biased between diagonal and adjacent neighbors:

Yang *et al.* [7] proposed image contrast levels based reversible visible watermarking method that embeds reconstruction information along with payload and thus does not require original watermark in order to remove it from image.

Chung *et al.* [8] have proposed reversible data hiding in quantized discrete cosine transform coefficients of image sequence. Yang and Tsai [9] have proposed use of interleaving predictions such that embedding is done in two stages where first odd columns are used for prediction of even columns and then even columns used for prediction of odd columns. Authors calculate prediction errors after predicting the pixel values and then use histogram shifting for embedding watermark data into these prediction errors. Liu and Tsai [10] have proposed one-to-one reversible compound mappings based visible reversible watermarking. These compound mappings are proved to be reversible mathematically and thus their reversibility is exploited to use them for reversible watermarking.

Chaumont and Puech [11] proposed dividing the differences into three categories namely embedding state, to correct state and original state. They embed the data only in the differences identified to be in embedding state. Hong and Chen [12] have used a variance control threshold to filter out high magnitude prediction errors from embedding process. They filter any prediction error from embedding if that prediction error's neighborhood has higher variance then the variance control threshold. Lin and Li [13] have proposed creating histograms of contiguous sub-portions of the image, rather than a single histogram of whole image. Histogram shifting is then used on this set of sub-image difference histograms that are placed in a tree for ordered representation. Luo *et al.* [14] have proposed to use difference histogram of high frequency sub-band of discrete wavelet transform for embedding data.

Yang *et al.* [15] propose to use histogram shifting on the histogram of differences of original pixel with median of its neighbors. This can be thought of as prediction error expansion with median of neighbors as predictor. Wang *et al.* [16] uses histogram shifting but rather than using inner region of histogram for embedding data they use the region emptied by shifting for placing the watermark data. Farrugia [17] propose a reversible visible watermarking algorithm for compressed images that embeds the residual information for image recovery along with the payload in the high frequency bands of quantized discrete cosine transform of image. Li [18] uses hamming code and distance between watermark data and the image. It uses bit negation to embed watermark in the image in the chosen bit for embedding.

Huang and Lin [19] have proposed prediction error expansion based method for medical images where they use their own proposed predictor that uses only three of the neighboring pixels.

In [20] difference expansion is used at block level such that differences of blocks rather than differences of pixels are used for embedding the watermark data.

Our research work is a prediction error expansion based reversible watermarking scheme. We have proposed four phase representation that allows non-overlapping representation of the image. At the same time the representation allows us to use a large prediction context for making prediction of image pixels. Our technique uses histogram shifting for expansion of prediction errors. The technique uses adaptive embedding and prediction error sorting to reduce image distortion for large and small payloads respectively. The combination of all these approaches became possible because of four phase representation that allows us to use sorting of prediction errors without compromising on size of prediction context. This has helped us improve the performance of our approach compared to other state of the art techniques in literature.

Li *et al.* [21] present a new prediction mechanism with improved embedding results for capacity embedding. A new idea of pixel value ordering is presented. The image is divided into non-overlapping blocks. Values of the pixel block are arranged in the ascending order. Pixels at the either end of the sorted sequence are modified. A maximum of 1,-1 change will be observed after the embedding scheme is applied. For an image of size $N \times M$ and for block size of $n \times m$, the methods produced a fairly large size location map. For each block 1 bit is required in location map. Size of the location map can be determined as NM/nm. The location map has to be compressed to increase space for data embedding. A maximum of 2 bits will be embedded in each block. By increasing the block size better imperceptibility can be achieved but it leads to less embedding capacity. For lena image of size 512 × 512 and a block size of 2 × 2 at most 32,000 bits can be embedded.

There are several research works in hybrid reversible watermarking schemes where prediction error expansion is either a part or the whole scheme or is inspired by that. Huang *et al.* [22] have proposed a hybrid of difference expansion and histogram shifting to achieve better capacity. Luo *et al.* [23] have proposed using interpolation in place of prediction or difference expansion. The variation does not exactly belong to prediction error expansion but is quite similar in the way the embedding is done. Kim *et al.* [24] have proposed using difference histograms of sub-sampled images rather than individual pixels for difference expansion. Tai *et al.* [25] have used the difference histogram for embedding the watermark, which is effectively the same as using difference expansion with histogram shifting.

III. PROPOSED METHOD

In this section we will explain proposed approach based on prediction error expansion for embedding reversible watermark. The proposed scheme is a four phase imperceptible watermarking scheme suitable for authentication of image content and authentication of source of origin.

The approach is based on decomposing the image into four non overlapping representations. Each of these representations is watermarked in a separate phase making a total of four phases for embedding a watermark. It is important to note that unlike the prediction schemes with overlapping prediction contexts, the order of scanning is not significant for our scheme. But for the matter of convention we scan each of the four phases in top-down and bottom-right fashion. In each scan the candidate pixel's for embedding are predicted using their prediction context and the prediction errors are calculated by comparing them with actual original values of the pixels.

Once the prediction errors are collected the histogram of these prediction errors is made and a bin range is selected to be watermarked. These pixels are further analyzed for overflow and underflow conditions. For those pixels that are likely to cause an overflow/underflow, an overflow table is created.

Algorithm 1 Proposed Encoding Algorithm

```
for k = 1 to 4 do
    for pixels in phase k do
        X′ = HybridPredictor(neighbors(X))
        PredictionError = X - X′
        T(i,1) = PredictionError
        T(i,2) = StandardDeviation(neighbors(X))
        i=i+1
    end
    Sort (T,2)
    (Tn,Tp) = CaculateHistogramBinRange()
    OverFlowMap = GenerateOverflowUnderflowMap()
    EmbedPayloadHeader(OverflowMap,Tn,Tp)
    for i = 1 to size(T) do
        if T(i,2) <= 1 then
            Expand prediction errors using equation (6)
        end
        else
            Expand prediction errors using equation (12)
        end
    end
end
```

Both the bin range for prediction error histogram bin and the payload size is needed by the decoder at the time of decoding. Thus the histogram bin range and the payload size are made a part of payload header. This is because the decoder will need to extract the header part to be able to use histogram shifting. However as you might have noticed header itself cannot be embedded using histogram shifting. This is because the decoder would not be able to use histogram shifting until it obtains the values of histogram bin range selection thresholds (T_n and T_p for negative and positive threshold respectively) and payload size. So the header is embedded by replacing the LSB of first 34 prediction errors rather than by histogram shifting. The first 34 prediction errors LSB also needs to be secured so that the watermarked image can be reversed back to the original image. Thus the LSBs of first 34 prediction errors are also included in the remaining payload for reversibility (detail will be discussed in the section on payload structure). Once the header is embedded in the image then histogram shifting is used to embed the least significant bits of header part, overflow table and user specified payload as watermark in the non-problematic (overflow/underflow) pixels of the image. This process is repeated for all four phases and the resultant image is reversibly watermarked.

Algorithm 2 Proposed Decoding Algorithm

```
for k = 1 to 4 do
    for pixels in phase k do
        X′ = HybridPredictor(neighbors(X))
        PredictionError = X - X′
        T(i,1) = PredictionError
        T(i,2) = StandardDeviation(neighbors(X))
        i=i+1
    end
    Sort (T,2)
    (Tn, Tp,OverflowMap) =
    ExtractInformationFromPayloadHeader()
    for i = 1 to size(T) do
        if T(i,2) <= 1 then
            Decode prediction errors using equation (11)
        end
        else
            Decode prediction errors using equation (13)
        end
    end
end
```

In the decoding stage the image is decomposed into four non-overlapping representations again. Similar to encoding stage each of the phase is scanned and the predictions are made for every pixel in the phase representation. After that the prediction errors are calculated and are sorted in the order of variance of the prediction context. The header part is then extracted from the LSB of first 34 prediction errors from the sorted list. This header part contains the histogram bin range and payload size for the embedded pixels. Once the histogram bin range and payload size are extracted the overflow map is calculated using the histogram bin shifting and double embedding technique (explained in detail in the histogram shifting section). To extract the remaining user specified payload histogram bin range and overflow map are used with histogram shifting. This process is repeated for each of the four phases. It is important to note that for decoding the phases are processed in inverse order from that of embedding.

The above discussion describes components interaction for our proposed watermarking scheme. Figure 1 summarizes this interaction in the form of a flowchart. In the following sections we will explain each component of the scheme individually in more detail.

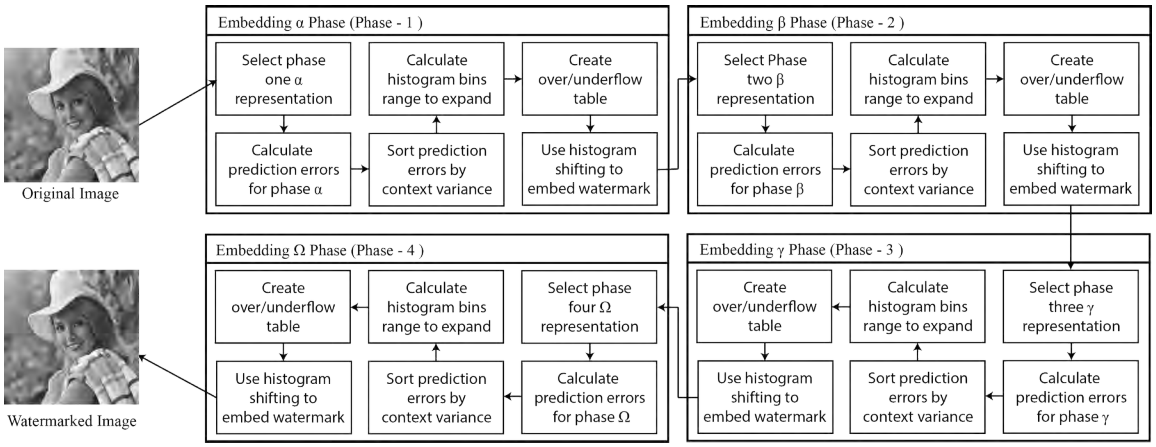

FIGURE 1. Four phase reversible watermark encoding flowchart.

A. FOUR PHASE DECOMPOSITION

Four phase decomposition is the decomposition of image into four non−overlapping set of pixels and prediction contexts. The significance of this decomposition is that in each individual phase there is no overlapping between the watermarked pixels and prediction context. There are two intents for using four phase representations rather than embedding the watermark in a single phase:

- We use sorting of prediction contexts as a mechanism for decreasing image distortion at smaller payloads. However, such a sorting needs to ensure that the same sorting order can be observed at both encoder and decoder. If the prediction context and watermarked pixels were to overlap with each other then the prediction context would change during encoding and decoding. Then we would not be able to get same sorted order on prediction contexts at encoder and decoder.
- By using four phases we can have a large neighborhood for each particular pixel. All 8 adjacent pixels of the candidate pixel are in the prediction context. A larger neighborhood provides more options for optimization of predictor by exploiting correlation with larger number of candidate pixels thus allowing for integration of more sophisticated predictors.

Four phase decomposition can be thought of as a sliding window of 3×3 sliding though the image with step size of 2 (jumping over a pixel at each step in both horizontal and vertical direction). Each phase has different starting pixel for the initial position of the sliding window. This sliding window starts with (2,2), (3,2), (2,3) and (3,3) for phases 1, 2, 3 and 4 respectively. This is represented in algorithm 3 below:

Such a division ensures that no pixel has its prediction context overlapping with any embedding candidate pixel. It is important to note that only an overlap between a prediction context and an embedding candidate pixel is a problem. Because prediction error sorting is done using variance of prediction context. Change in prediction context would might cause different variances of prediction contexts. Such a difference of variance in encoding and decoding stage will reveal a different sorting order at encode and decode stage. Thus, the prediction contexts will not be changed and only prediction candidate pixels are modified. An overlap between the two prediction context is not a problem as neither will change during embedding. Figure 2 describes the representations created during each of the four phases.

B. HYBRID PREDICTOR

Predictor selection is very significant choice in a prediction error expansion scheme as the accuracy of predictor contributes to reduction of distortion in the image. We have empirically observed image data split on the basis of its variance using several different predictors. Based on these results we have built our hybrid predictor which uses three different prediction schemes for different neighborhood variance ranges.

Three ranges are marked on which three different predictors are used. Low variance range consists of all prediction contexts whose variance is in the range of 0 to 10. Medium variance range is for variance of 10 to 40 whereas high variance range include all prediction contexts whose variance is greater than 40. For low variance range mean predictor is used. Intuition is that the prediction candidate value is very likely to be similar to the majority of neighboring pixels for lower variance. For medium variance range trimmed mean predictor is used. The intuition is to trim off the outliers that are very likely to occur in the medium variance range. Because in medium variance range most values are similar but some outliers take the variance value into medium range. High variance values in natural images are mostly an indication of edges and sharp regions. MED (predictor from jpeg standard) predictor is likely to perform good on edge and sharp regions. However for high variance range trimmed mean of four sided MED predictor is used rather than one sided MED to reduce the impact of tainted neighborhood which occurs during watermarking. The neighborhood is tainted in all phases after first phase. By tainted we mean

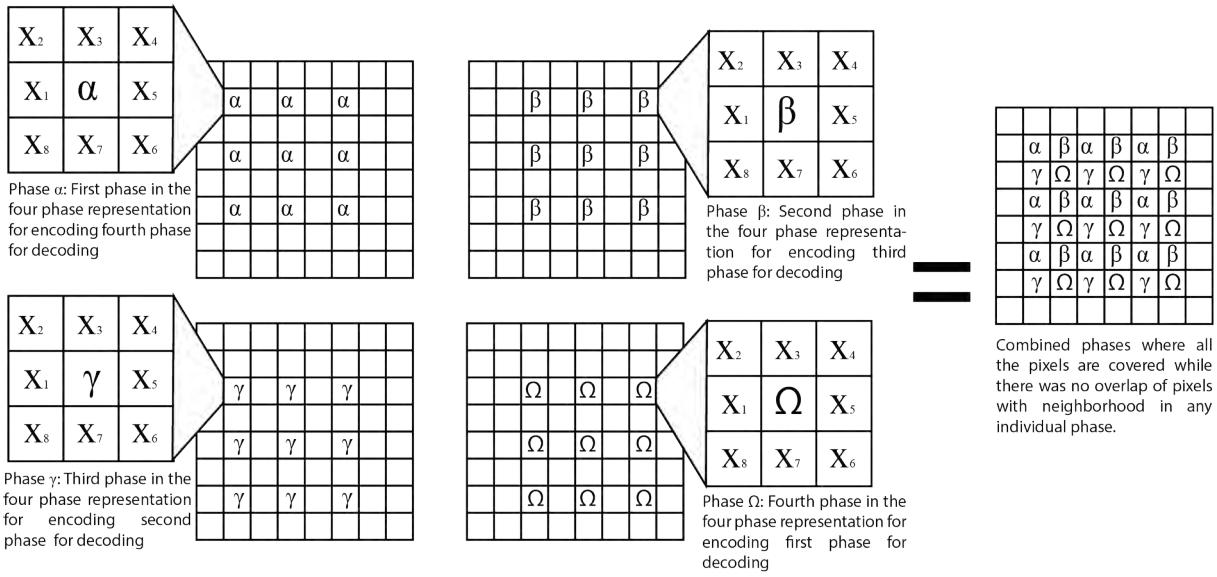

FIGURE 2. Four phase representation of the image where there is no overlap between pixels and neighbors of a single phase. α, β, γ and Ω represents first, second, third and fourth phase respectively.

all phases other than first phase will have prediction context of the pixels modified because of prediction error expansion. The predictor is defined using the following equation:

$$X' = \begin{cases} \sum_{i=1}^{8} \frac{X_i}{8} & \text{if } \sigma_x \geq 0 \text{ and } \sigma_x < 10 \\ \sum_{i=3}^{6} \frac{S_i}{4} & \text{if } \sigma_x \geq 10 \text{ and } \sigma_x < 40 \\ \sum_{i=2}^{3} \frac{SM_i}{2} & \text{if } \sigma_x > 40 \end{cases} \quad (1)$$

Where X' is predicted pixel value, X is the collection of eight neighboring pixels, S is the collection of eight neighboring pixels sorted in ascending order and SM is the collection of four side's MED predicted values sorted in ascending order. First condition triggers the mean predictor that represents mean of 8 neighboring pixels. Second condition triggers trimmed mean that represents mean of middle four elements of sorted 8 neighboring pixels. First the 8 neighbors of the pixel are sorted. Next only four middle values are selected amongst the sorted neighbors. Mean of these four middle values amongst the sorted neighbor is considered the prediction value. The last condition triggers the trimmed mean of four sided MED predictor. Because the predictor is based on JPEG standard's MED predictor we will first explain MED predictor. MED predictor is defined as:

$$X = \begin{cases} min(A, B) & \text{if } C \geq max(A, B) \\ max(A, B) & \text{if } C \leq min(A, B) \\ A + B - C & otherwise \end{cases} \quad (2)$$

In our four sided MED predictor we use all corners for predicting the value. There are four corners when using prediction context of eight surrounding neighbors. So we get four MED predictor based predictions:

1) A = Left neighbor, B = Top neighbor, C = Top-Left neighbor
2) A = Top neighbor, B = Right neighbor, C = Top-Right neighbor
3) A = Right neighbor, B = Bottom neighbor, C = Bottom-Right neighbor
4) A = Bottom neighbor, B = Left neighbor, C = Bottom-Left neighbor

We sort these predictions and call the resulting collection SM that contains the four predicted values, one for each of the corners on which MED predictor is applied. Then the trimmed mean (only middle two value's mean) is taken as the predicted value, this enables reduction of noise due to tainted neighbor during watermarking. By tainted neighbor we mean that in all phases of watermarking other than the first phase prediction context is modified for embedding data. With the prediction context changed/tainted it become even harder to predict an accurate value without trimming outliers.

C. SORTING FOR PIXEL SELECTION

For smaller payloads if we can sort the pixels such that we select the subset of candidate pixels with smallest prediction error for watermark embedding we can achieve minimum image distortion. However, we know that prediction error depends on both the pixel's value and its predicted value. Thus because the pixel value is different in embedding and extraction phase we cannot have the same prediction errors for same pixels in embedding and extraction. This means that we would not be able to recognize the order in which the watermark bits were embedded. So sorting on the basis of prediction error is highly desirable but not possible.

However, we can sort with good efficiency even if we can *estimate* the relative magnitude of prediction errors. We know

Algorithm 3 Algorithm for Four Phase Representation Using Sliding Window

```
// Phase 1:
for i=2 to width-1 step 2 do
    for j=2 to height-1 step 2 do
        Prediction context = {Image(i,j-1),
        Image(i-1,j-1),Image(i-1,j),Image(i-
        1,j+1),Image(i,j+1),Image(i+1,j+1),Image(i,j),
        image(i,j-1)};
        Prediction candidate = Image(i,j);
    end
end
// Phase 2:
for i=2 to width-1 step 2 do
    for j=3 to height-1 step 2 do
        Prediction context = {Image(i,j-1),
        Image(i-1,j-1),Image(i-1,j),Image(i-
        1,j+1),Image(i,j+1),Image(i+1,j+1),Image(i,j),
        image(i,j-1)};
        Prediction candidate = Image(i,j);
    end
end
// Phase 3:
for i=3 to width-1 step 2 do
    for j=2 to height-1 step 2 do
        Prediction context = {Image(i,j-1),
        Image(i-1,j-1),Image(i-1,j),Image(i-
        1,j+1),Image(i,j+1),Image(i+1,j+1),Image(i,j),
        image(i,j-1)};
        Prediction candidate = Image(i,j);
    end
end
// Phase 4:
for i=3 to width-1 step 2 do
    for j=3 to height-1 step 2 do
        Prediction context = {Image(i,j-1),
        Image(i-1,j-1),Image(i-1,j),Image(i-
        1,j+1),Image(i,j+1),Image(i+1,j+1),Image(i,j),
        image(i,j-1)};
        Prediction candidate = Image(i,j);
    end
end
```

that the pixels in an image are highly correlated with their neighboring pixels (in our case the prediction context). This is because predictor's accuracy is dependent on how much the neighboring pixels are correlated with the predicted value. Keeping this in view we estimate the prediction error by estimating the variance between the 8 neighboring pixels i.e. prediction context:

$$\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu) \tag{3}$$

In the above equation N is 8 indicating the count of neighbors, x_i represent each of eight neighbors in the prediction context and μ is arithmetic mean of eight neighbors. Variance of prediction context gives us a good estimate of relative magnitude of prediction errors. If we sort the estimated prediction errors relative magnitude, we get a sorting order of bits which closely resembles sorting with respect to actual prediction errors.

Figure 3 shows correlation between prediction errors and variance of prediction context. The plotted values of prediction errors are sorted by the variance of their prediction context. Taking this as a pretext we use variance of prediction context for pixel selection which gives very close estimate of prediction errors while at the same time remains unchanged during encoding and decoding for each phase.

D. HISTOGRAM BIN RANGE

We know that the magnitude of distortion caused by expanding a prediction error is directly proportional to magnitude of prediction error itself. Keeping this in view we select range of histogram bins for watermark embedding such that our selection contains only the lowest magnitude bins. For our histogram bin range selection we calculate two thresholds T_n and T_p that represent negative and positive threshold respectively. T_n and T_p are thresholds that encloses the region in the prediction error histogram that contains the bins to be expanded.

We start by including histogram bin for prediction error zero in our bin selection range by setting T_n and T_p to zero. The magnitude of the histogram bin (not the magnitude of prediction error) determines how many bits the bin can hold. For example, if the magnitude for histogram bin for prediction error zero is 100 it contains 100 prediction errors so all of them can be expanded to hold up to 100 bits of data (assuming no underflow/overflow). We use this assumption to expand the histogram bin range iteratively until it is sufficient to contain all the data bits.

Algorithm 4 Adjusting Histogram Bin Range for Embedding

```
step = Even;
While ∑_{Tn}^{Tp} HistogramBins < Payload Size
if Even then
    T_n = T_n - 1;
else
    T_p = T_p + 1;
end
```

In every iteration of expansion we compare the size of payload with sum of magnitudes of selected bin range bins. If the payload fits in the given bin range we use T_n and T_p as thresholds and break. Otherwise we modify T_n and T_p one by one in each iteration. On first iteration we move the T_n threshold to left by subtracting one from it. If the capacity is still not sufficient to contain all the payload bits we move threshold T_p to the right by adding one to it. Similarly keep on incrementing T_p and decrementing T_n turn by turn until the capacity is sufficient to hold complete payload.

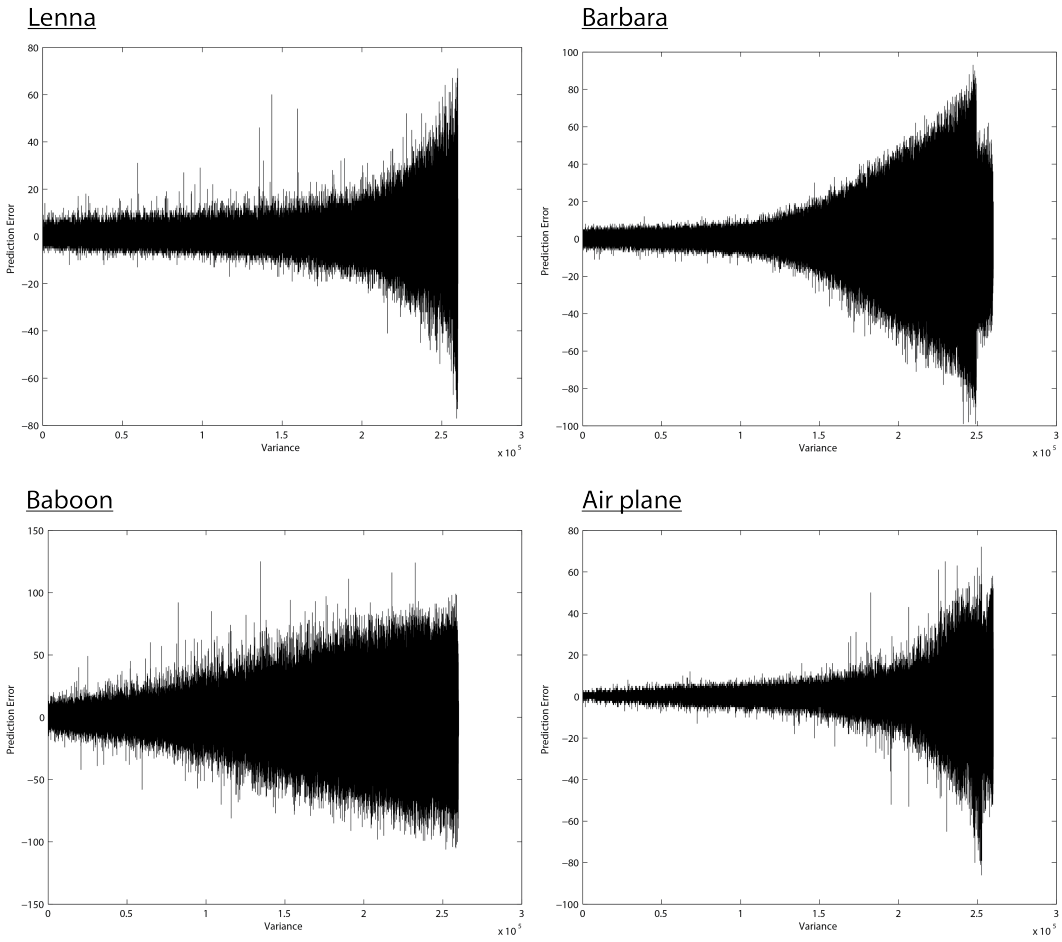

FIGURE 3. Prediction errors ordered using sorting by variance, showing the high correlation between prediction errors magnitude with variance.

One exception is the case in which the payload size is smaller than magnitude of zero prediction error histogram bin. For example, there are only 100 or less data bits to be embedded and we have 100 or more prediction errors in histogram bin of all those prediction errors that have zero magnitude. In such a scenario, we keep both T_n and T_p at zero. It is important to note that this is only initial threshold calculation and may need to be changed if overflows/underflows are detected. This is discussed in next section.

E. OVERFLOW/UNDERFLOW TABLE

Once we have selected a histogram bin range for embedding the watermark we need to find the problematic pixels in the image which cannot hold a data bit. It may be possible that a bin has small prediction errors but expanding that prediction error can cause an overflow or underflow condition. This means that once we expand the prediction error and add them to the predicted value there is a possibility that it can overflow i.e. the value exceeds 255 or underflow i.e. the value becomes less than 0.

For example, assume we have a pixel value 255. Predictor is applied on the prediction context and that gives prediction accurately as 255. The prediction error is 0 which we expand to hold bit value 1. So the expanded prediction error becomes 1 while the predicted value is 255. If we add them together they will become 256 which is an overflow condition if one byte is used for representing a pixel. This is because the value 256 cannot be stored in the byte that represents pixel value.

There is a problem with leaving some data bits without embedding. The problem is that how would the decoder be able to identify whether or not encoder embedded a data bit in the given prediction error. Thus the decoder needs to be explicitly informed about the pixels which contain embedded data and the ones that do not. To communicate this information to decoder, several research works use mapping table. The idea is to put 1 for pixels that contain data and 0 for those that does not contain data in a linear table that is embedded in the image along with the actual user specified payload. However, size of such a mapping table is equal to number of pixels available for embedding (as all of them may or may not have contained data bit). Most of techniques in literature relies on compression of mapping table for creating space for data bits. One of the main challenges of difference expansion

and prediction error expansion has been to produce a smaller representation of mapping table.

For our research work mapping table is represented by an overflow table representation inspired from an idea in the field of difference expansion proposed by Kim *et al.* [26]. Unlike most other research works Kim *et al.* [26] create a smaller representation of the mapping table rather than merely compressing it. We discuss this idea in the next two subsections for encoding and decoding stage respectively.

1) CREATING AND EMBEDDING OVERFLOW TABLE IN ENCODER

In the embedding phase the technique relies on reducing the number of entries to maintain overflow information for expandable bits. The idea is to try to expand each of the candidate prediction error twice with hard bit. Hard bit is bit 1 for positive prediction errors and bit 0 for negative prediction errors, because they are more likely to cause an overflow. After the candidate prediction is expanded twice we categorize it by whether the pixel can be expanded once, twice or never without overflow/underflow. This provides us with three cases:

1) Non-Expandable i.e. expanding the prediction error even once causes overflow/underflow. We cannot expand these prediction errors so we do not expand them and keep an entry in the overflow table.
2) Ambiguously-Expandable i.e. expanding the prediction error second time causes overflow/underflow. We expand the prediction error but keep an entry in the overflow table.
3) Unambiguously-Expandable i.e. prediction error does not cause overflow/underflow in either of the two expansions. We expand the prediction error and do not need to keep any entry in the overflow table.

A traditional mapping table contains one entry for each of the embedding candidates prediction error whereas the described overflow/underflow map will not need entries for unambiguously expandable prediction errors. We still need entries for ambiguously expandable and non-expandable pixels. However, in natural images the ambiguously expandable prediction errors and non-expandable prediction errors are much fewer in number compared to number of unambiguously expandable prediction errors. This overflow/underflow map representation of overflow map reduces table size considerably and therefore helps in reducing image distortion. For more descriptive pictorial example see figure 4.

2) ACQUIRING AND USING OVERFLOW TABLE IN ENCODER

Decoder constructs the overflow table in similar manner to encoder. Once it has extracted the histogram bin range thresholds it expands pixels in that range *once*. This single expansion attempt by decoder is equivalent to second attempt of encoder in the process of embedding watermark (as the data that decoder received is already expanded once by the encoder). If there is no overflow then the decoder assumes the prediction error to contain a data bit and it being unambiguously expandable. Otherwise it uses the mapping table. We have a total of three cases for extraction:

1) Those prediction errors that do not overflow or underflow on expansion are determined as unambiguously expandable. Data bits are extracted from these prediction errors.
2) Those prediction errors that get overflow are not unambiguously expandable. Therefore we look at the mapping table at the current index. If there is a 1 in the mapping table then it is determined as ambiguously-expandable. *The data bits are extracted from these prediction errors.* Also the current index in the mapping table is incremented by one.
3) Those prediction errors that overflow underflow and when we look at the mapping table at the current index we find a 0 entry then the prediction error is determined as Non Expandable. These prediction errors were not expanded and thus no data is extracted from these prediction errors.

Lets briefly summarize overflow/underflow table section in the overall embedding process. Once we have made the pixel selection we create an overflow table as described in this section. The mapping table also becomes a part of payload that is embedded in the image as we will see during the embedding process in the next section.

F. HISTOGRAM BIN SHIFTING

Once the proposed scheme has selected prediction errors which need to be expanded, we need an efficient way to expand the selected prediction errors. For expansion of prediction errors we use histogram shifting. Histogram shifting is capable of expanding prediction errors at lower distortion while avoiding the overlap problem. Histogram shifting in our paper is inspired from prediction error expansion technique originally proposed by Thodi and Rodríguez [4]. In traditional histogram shifting techniques grey level histogram of the image is considered such as in [27]. However, when used with prediction error expansion techniques the histogram of prediction errors is used rather than grey level histogram of image. Basic idea is to avoid the overlap caused by expanding the prediction errors in the selected range. All prediction errors other than those in the selected range are shifted to avoid their overlap with the expanded pixels.

G. ENCODING USING HISTOGRAM BIN SHIFTING

For histogram shifting encoder had previously determined parameters T_n and T_p. The encoder now tries to expand the prediction errors in the range between T_n and T_p. For example, if negative threshold T_n is -2 and positive threshold T_p is 2 then we would like to expand all the prediction errors of magnitudes -2, -1, 0, 1 and 2. Expanding comprises of shifting the prediction error value left by one bit and then adding the data bit. This expansion can be represented as:

$$P'_{i,j} = 2P_{i,j} + b \tag{4}$$

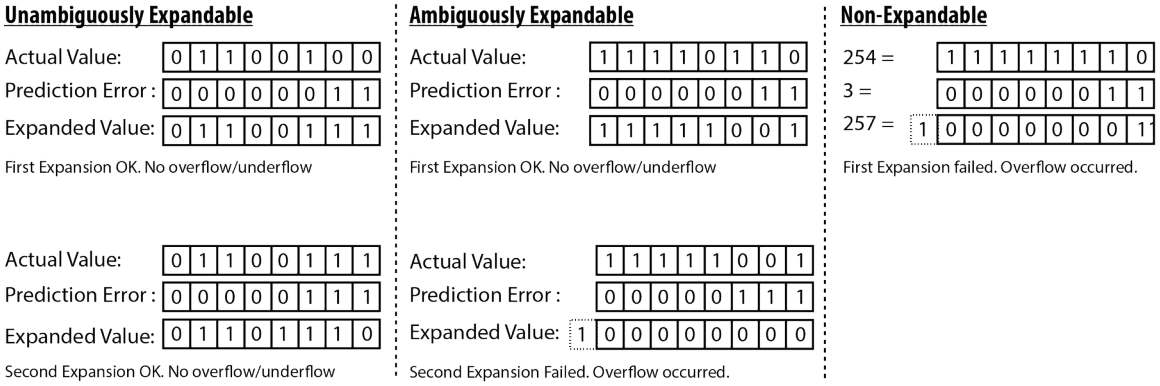

FIGURE 4. Double expansion for overflow map creation to determine unambiguously expandable, ambiguously expandable and non-expandable prediction errors.

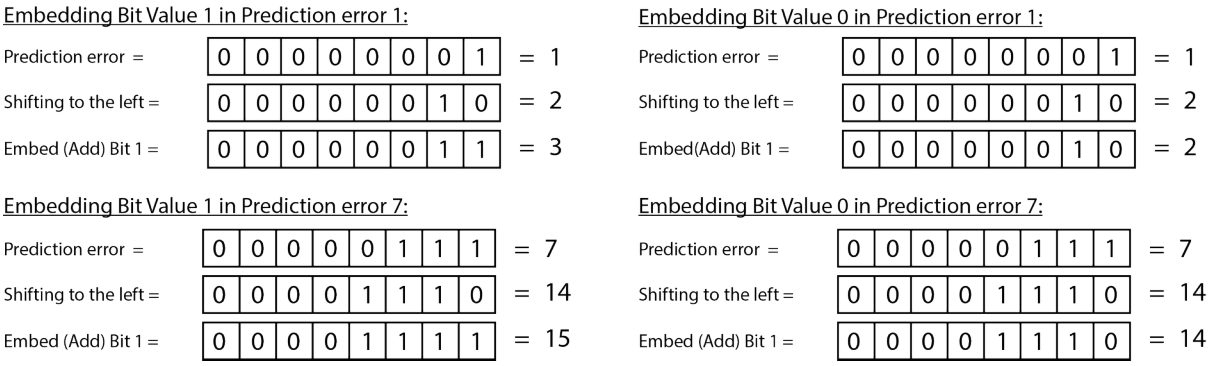

FIGURE 5. Process of bit expansion using prediction error expansion on prediction errors of two different magnitudes. Viewer may also observe that magnitude of distortion directly proportional to magnitude of prediction error.

In the above equation P, b and P' represent prediction error, data bit and the watermark embedded prediction error respectively. The expansion process is also pictorially described in figure 5. It demonstrates expansion of two different prediction errors. It can also be

observed that larger prediction errors produce large distortion on expansion. To explain the process of expansion lets consider the prediction error at row pixel on 7 and column 8 we have a prediction error P of 1 and we want to embed bit value 1 in it. The result after expansion would be 3 as solved below:

$$P'_{7,8} = 2P_{7,8} + b = (2 \times 1) + 1 = 3 \tag{5}$$

However, there is one problem that must be addressed for this approach to work. The problem is of overlap between expanded values and non-expanded values. When proposed scheme expands the prediction errors they become members of another histogram bin. For example, lets take one scenario as the equation above and consider yet another scenario where we have prediction error of 3 at row 9 and column 9 (i.e. P(9,9) of image). Now we have two prediction errors with magnitude 3 in the watermarked image.

The problem is that decoder must be able to identify which one of them contains the embedded bit and which one was 3 before embedding (as 3 won't be expanded as it doesn't belong to range (T_n, T_p)). This overlap happens because proposed method has actually shifted the histogram for expanding the prediction errors, which takes the expanded prediction errors into range $(2T_n, 2T_p + 1)$. The overlapping problem is solved by shifting the histogram of all the remaining prediction errors that does not belong to (T_n, T_p). To prevent overlap the proposed approach shifts the prediction errors not belonging to (T_n, T_p) by T_n for the negative prediction errors and by $T_p + 1$ for positive prediction errors.

With the above shifting scheme and prediction error expansion our complete equation for watermark embedding becomes:

$$P'_{i,j} = \begin{cases} 2P_{i,j} + b & if P_{i,j} \in [Tn; Tp] \\ P_{i,j} + T_p + 1 & if P_{i,j} > T_p \; and \; T_p \geq 0 \\ P_{i,j} + Tn & if P_{i,j} < T_n \; and \; T_n < 0 \end{cases} \tag{6}$$

The equation is pictorially explained in Figure 5. The prediction errors are shifted to make space for the expanded prediction errors. Negative errors are shifted leftwards whereas positive errors are shifted rightwards to create space for the expanded prediction errors.

Let us now consider example above to explain how the overlap will be resolved. Proposed scheme had an expanded prediction error in histogram bin 1 overlapping with a non expanded prediction error in histogram bin 3. The expanded value will still remain 3 as the equation for expansion remains the same. However the non-expanded value 3 would be

shifted using the shifting scheme described above to resolve overlap:

$$P'_{i,j} = P_{i,j} + T_p + 1 = 3 + 2 + 1 = 6 \tag{7}$$

This shift will take the non-expanded prediction error out of the overlap range $(2T_n, 2T_p + 1) = (-4, 4)$ in the given example. This allows for resolution of overlap region without a need to have a dedicated data structure that contrasts expanded prediction errors from non-expanded prediction errors in the overlap range.

H. DECODING USING HISTOGRAM BIN SHIFTING

In reversible watermarking extraction decoder has to achieve two aims namely recovery of correct data and recovery of zero error original image. When extraction phase reaches histogram shifting for extracting watermark it already has extracted the histogram range thresholds (T_n, T_p) and the overflow table. The extraction scheme can be divided into two parts. First part focuses on recovery of the embedded data and second part focuses on recovery of zero error original image.

For first part of extraction namely recovery of embedded data proposed method scans for all the prediction errors that belong in the range $(2T_n, 2T_p + 1)$. For each of these prediction errors, we know that they will either contain a data bit or contain a mapping table entry. We have already determined whether or not the prediction error contains the data bit in section on overflow table. So for the prediction errors belonging to ambiguously-expandable and unambiguously-expandable category the proposed approach extracts the data bits. To extract the data bits, the proposed approach takes the remainder of embedded prediction error with 2. This can be represented as:

$$b = P'_{i,j} \; mod \; 2, \quad \forall P_{i,j} \in [2T_n; 2T_p + 1] \tag{8}$$

For the second part of decoding namely recovery of zero-error original image proposed approach need to revert the expanded prediction errors back to their original values. This is done by dividing the predicted error by 2, this shifts back the prediction error back to its original value. Then taking the floor makes sure that if the embedded bit was 1 its half impact of 0.5 is also removed from the original prediction error value. This can be represented as:

$$P_{i,j} = \left\lfloor P'_{i,j}/2 \right\rfloor \tag{9}$$

Once we have the original prediction error, proposed method adds it to the predicted value determined using the neighboring prediction context. Adding the original prediction error to predicted value proposed method gets the zero-error original pixel value. This can be represented as:

$$I_{i,j} = P_{i,j} + I'_{i,j} \tag{10}$$

Where $I_{i,j}$, $I'_{i,j}$ and $P_{i,j}$ represent zero-error original image pixel, predicted image pixel value, and zero-error original prediction error respectively.

Algorithm 5 Adaptive Embedding

if *Variance* ≤ 1 *and bpp* > 0.5 **then**
  Embed two bits using equation (12)
end
else
  Embed one bit using equation (6)
end

Proposed approach also needs to shift prediction errors back to their original values, which were earlier shifted to avoid overlap with expanded pixels. This shifting is required to recover the original image, these shifted prediction errors do not contain any data. The shifted values are all those values that are either less than $2T_n$ or greater than $2T_p + 1$. These values are only shifted in the same magnitude but opposite direction as in embedding, to reverse the impact of shifting. The values greater than $2T_p + 1$ are shifted $T_p + 1$ leftwards by subtracting $(T_p + 1)$ from the prediction errors. Similarly prediction errors less than $2T_n$ are shifted rightwards by subtracting (T_n). The complete shifting and extraction can be represented as under:

$$P_{i,j} = \begin{cases} \left\lfloor P'_{i,j}/2 \right\rfloor & if P'_{i,j} \in [2T_n, 2T_p + 1] \\ P'_{i,j} - T_p - 1 & if P'_{i,j} > 2T_p + 1 \;\; and \;\; T_p \geq 0 \\ P'_{i,j} - T_n & if P'_{i,j} < 2T_n \;\; and \;\; T_n < 0 \end{cases} \tag{11}$$

I. ADAPTIVE EMBEDDING

When a prediction scheme is used for embedding at higher capacities it is likely to encounter both high variance and low variance prediction contexts. In the higher variance contexts the prediction accuracy is likely to be degraded and cause larger distortion in the watermarked image. Proposed scheme uses adaptive embedding based on variance of the prediction context to control this phenomena. Proposed approach embeds two bits in low variance regions with variance less than or equal to 1.0 and for higher bpp payloads of bpp of more than 0.5.

The idea is to embed two bits in low variance smooth regions while only one bit in high variance regions when embedding at higher capacities. However expansion of prediction errors also calls for change in shifting the remaining pixels accordingly as to avoid overlap in the expanded and non expanded values. For embedding two bits following equation is used:

$$P'_{i,j} = \begin{cases} 4P_{i,j} + 2b_1 + b_2 & if P_{i,j} \in [T_n; T_p] \\ P_{i,j} + 3T_p + 3 & if P_{i,j} > T_p \; and \; T_p \geq 0 \\ P_{i,j} + 3T_n & if P_{i,j} < T_n \; and \; T_n < 0 \end{cases} \tag{12}$$

It can be observed from above representation that for embedding data bits the prediction is shifted by two bits. After the shifting the data bits are added to it. All the values not in the selected histogram bin range need to be shifted. However this time they are shifted more as the overlap region increases

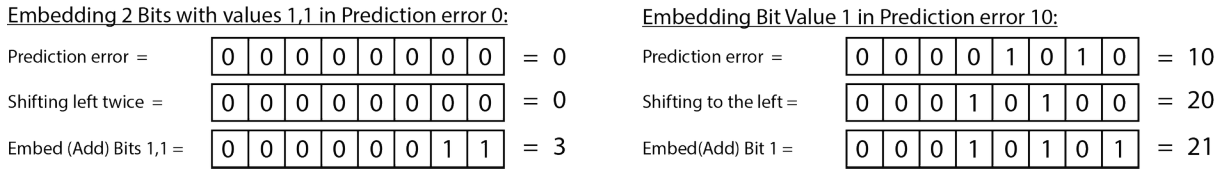

FIGURE 6. Expanding larger prediction error once can cause more distortion than expanding smaller prediction errors twice.

Algorithm 6 Timing Analysis of Encoding Algorithm

Legend:
N = (ImageWidth-2) × (ImageHeight-2)
Complexity Analysis:
for *k = 1 to 4* **do**
  for *pixels in phase k* **do**
    X' = HybridPredictor(neighbors(X)) ⟵ O(C)
    PredictionError = X - X' ⟵ O(C)
    T(i,1) = PredictionError ⟵ O(C)
    T(i,2) = StandardDeviation(neighbors(X)) ⟵ O(C)
    i=i+1 ⟵ O(C)
  end
  Sort (T,2) ⟵ O(1/4 (N lg N))
  (T_n,T_p) = CaculateHistogramBinRange() ⟵ O(C)
  OverFlowMap = GenerateOverflowUnderflowMap() ⟵ O(N/4)
  EmbedPayloadHeader(OverflowMap,T_n,T_p) ⟵ O(C)
  for *i = 1 to size(T)* ⟵ *O(N/4)* **do**
    if *T(i,2) <= 1* **then**
      Expand prediction errors using equation (6) ⟵ O(C)
    end
    else
      Expand prediction errors using equation (12) ⟵ O(C)
    end
  end
end
Time Complexity:
N lg N

Algorithm 7 Timing Analysis of Decoding Algorithm

Legend:
N = (ImageWidth-2) × (ImageHeight-2)
Complexity Analysis:
for *k = 1 to 4* **do**
  for *pixels in phase k* **do**
    X' = HybridPredictor(neighbors(X)) ⟵ O(C)
    PredictionError = X - X' ⟵ O(C)
    T(i,1) = PredictionError ⟵ O(C)
    T(i,2) = StandardDeviation(neighbors(X)) ⟵ O(C)
    i=i+1 ⟵ O(C)
  end
  Sort (T,2) ⟵ O(1/4(N lg N))
  (T_n, T_p,OverflowMap) = ExtractInformationFromPayloadHeader() ⟵ O(C)
  for *i = 1 to size(T)* **do**
    if *T(i,2) <= 1* **then**
      Decode prediction errors using equation (11) ⟵ O(C)
    end
    else
      Decode prediction errors using equation (13) ⟵ O(C)
    end
  end
end
Time Complexity:
N lg N

due to increased change in the expanded values. Positive prediction errors not in the histogram bin range are shifted by $3T_p + 3$ which takes them out of expanded prediction errors range. Similarly negative prediction errors are shifted by $3T_n$ that takes them out of expanded prediction errors range.

Decoding also needs to be adjusted for adaptive embedding when extracting two bits, this is represented by the following equation:

$$P_{i,j} = \begin{cases} \left\lfloor P'_{i,j}/4 \right\rfloor & if P'_{i,j} \in [4T_n, 4T_p + 3] \\ P'_{i,j} - 3T_p - 3 & if P'_{i,j} > 4T_p + 3 \ \ and \ \ T_p \geq 0 \\ P'_{i,j} - 3T_n & if P'_{i,j} < 4T_n \ \ and \ \ T_n < 0 \end{cases} \tag{13}$$

The range of prediction errors for low variance prediction errors which are considered expanded is increased to $[4T_n, 4T_p+3]$ because of the change in embedding. Similarly the shifting is reverted accordingly for negative and positive prediction errors to acquire the same image as the one before embedding.

Above equation helps in acquiring the original image back while the two data bits are extracted using the following equation:

$$\begin{aligned} b_1 &= \left\lfloor (P' - (4 \times \lfloor P'/4 \rfloor))/2 \right\rfloor; \\ b_2 &= P' - 2 \times \lfloor P'/2 \rfloor \end{aligned} \tag{14}$$

Primary intuition behind adaptive embedding is the fact that embedding two bits in fairly smooth region causes less distortion than embedding even a single bit in very high frequency (variance) regions as shown in figure (6). So by adaptive embedding proposed approach tries to embed more data in the smaller variance region as to avoid the need to

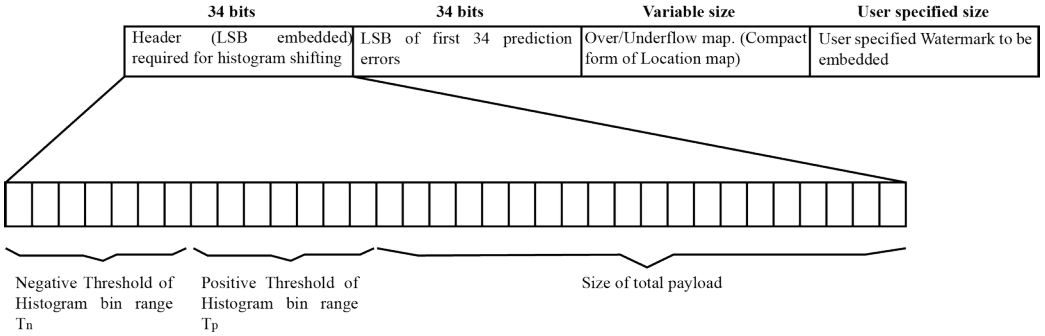

FIGURE 7. Payload structure to be embedded as watermark in image.

embed in the larger variance regions thus resulting in reduction of image distortion.

J. PAYLOAD STRUCTURE

Proposed approach uses histogram shifting for embedding data bits that allowed us to avoid overlapping problem. However, histogram shifting needs some extra data to be able to extract data. For instance it is impossible to use histogram shifting for data extraction unless the positive threshold T_p, negative threshold T_n and payload size are known. This essential extra information is called payload header and must also be embedded in the image along with the user specified payload. However, as histogram shifting cannot be used to extract the header part (because the essential parameters aren't known at the time of extraction of these parameters themselves), this header part needs to be stored such that it can be extracted without histogram shifting.

The idea is to use least significant bit replacing scheme for first 34 sorted prediction errors. However, bit replacement will change the values stored in the first 34 prediction errors irreversibly. In order to ensure reversibility proposed method stores the original LSBs of the first 34 prediction errors as part of payload.

Overflow/underflow map is also essential for extraction of data bits using histogram shifting thus it also becomes a part of payload header. So the complete header include negative threshold T_n, positive threshold T_p, payload size, LSB of first 34 prediction errors and the overflow/underflow map. Figure 7 represents the complete structure of the payload. This payload header is embedded in the image along with user specified payload to ensure self dependency of the image without requirement of any side channel for transfer of any extra information.

IV. EXPERIMENTAL RESULTS

In this section we will discuss the results of experimentation and compare them with some of existing state-of-the-art techniques in the literature. As a comparison metrics, we use watermarked image distortion at a given capacity as the performance measure for reversible watermarking scheme. Lower distortion indicates good performance of a watermarking scheme. For this purpose we use PSNR as our distortion measure that reflects the amount of distortion introduced by embedding watermark. Now we will discuss the results of experimentation and the performance measure used for reporting results of experimentation.

A. DISTORTION MEASURE

We use peak signal to noise ratio PSNR as distortion measure. PSNR is defined based upon mean square error(MSE) of the watermarked image with original image. MSE is calculated as:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{15}$$

Where I and K are original and modified images respectively. Whereas m and n are width and height of the images respectively. PSNR is then calculated using:

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{16}$$

Where MAX_I^2 is the square of maximum possible value in an image. This is generally 255 for gray scale images. MSE is the mean square error calculated in the above equation. A reversible watermarking technique is considered good if its watermarked image gives high PSNR (less distortion) with the original image at a given capacity as compared to other techniques.

Reason for using Peak Signal to Noise Ratio (PSNR) rather than Mean Square Error (MSE) is that most of the watermarking research community uses PSNR for reporting their results. Therefore in order to compare the results of experimentation with other techniques in literature PSNR is used as performance measure.

B. TIME COMPLEXITY ANALYSIS

An important factor for measuring performance of a watermarking scheme is the time required for embedding and extracting watermark. In order to gauge the performance measure we analyze the order of growth of the algorithm. Both encoding and decoding have worse case order of growth of O(N lg N). Where N is all pixels in the image except for boundary pixels of the image. The count of these pixels can

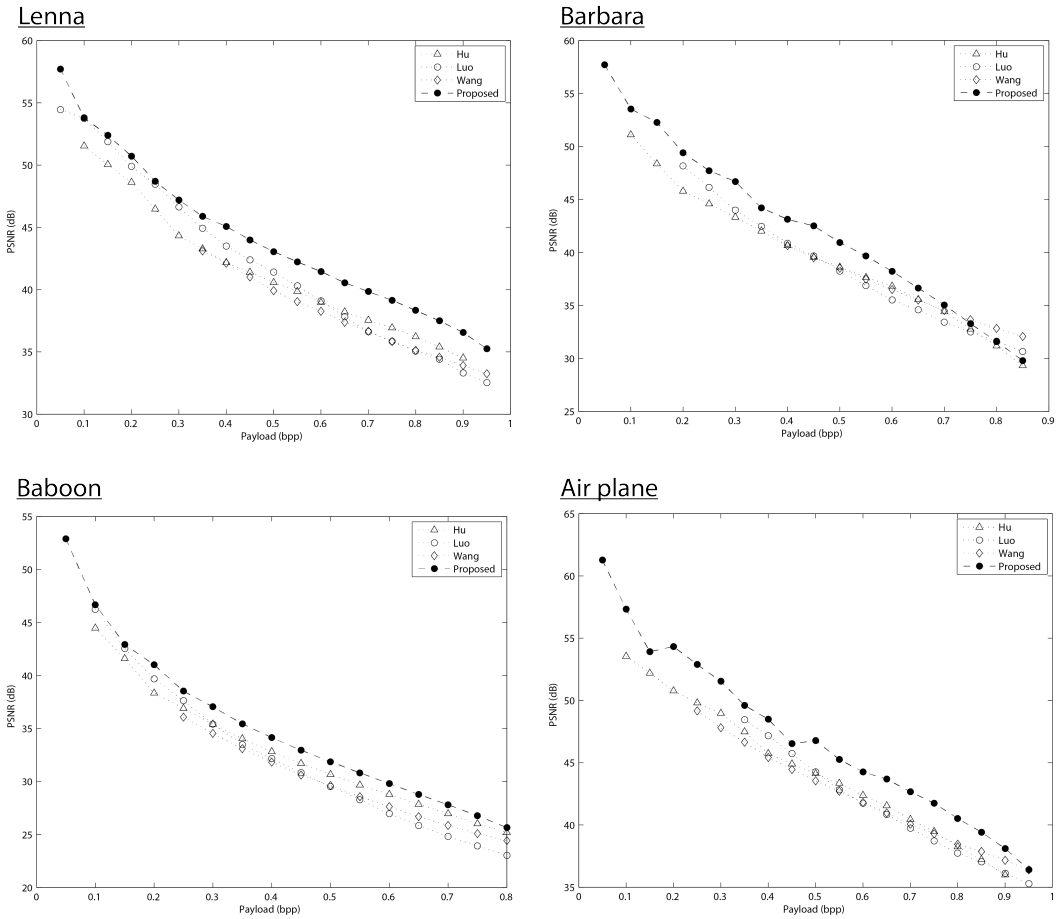

FIGURE 8. Results compared with existing state of the art techniques for reversible watermarking.

be represented by (ImageWidth-2) × (ImageHeight-2). This shows that the technique has reasonable fast speed and is suitable for reversible watermarking in real world applications.

C. DISCUSSION

Experiments were performed to evaluate performance of proposed approach in terms of low distortion in the watermarked image. In this study four standard 512 × 512 images Lenna, Baboon, Barbara and Airplane images are being used to compare our results with some state of the art techniques in the literature. On x-axis we have payload size in units of bits per pixel (bpp) (also called embedding rate). Whereas on y-axis we have Peak Signal to Noise Ratio (PSNR) that measures image quality (higher PSNR indicates lower distortion in the watermarked image). We consider an embedding scheme *A* superior to embedding scheme *B* if embedding scheme *A* embeds a payload of same size at a lower distortion than embedding scheme *B*.

Results are compared with prediction error expansion based method of Hu *et al.* [28], interpolation-error expansion based method of Luo *et al.* [23] and integer transform based method of Wang *et al.* [20]. Reason for selecting these three research works is because they cover a vast range of techniques for reversible watermarking. Hu *et al.* [28] method represents prediction error expansion methods. Whereas Luo *et al.* [23] method uses interpolation thus the prediction is dependent on larger prediction/interpolation context. Comparison with this paper provides diversity as to analyze how a completely different paradigm performs on the same images. Similarly integer transform based method of Wang *et al.* [20] is the state of art technique yet different from regular prediction error expansion. Thus together these three techniques represent three state of the art classes of reversible watermarking techniques.

Results of experimentation are shown in Figure 8. For Lena image, proposed scheme outperforms other three competing techniques [20], [23], [28] by causing less distortion in the watermarked image. Similar results can be observed for Baboon and Airplane Images where the proposed scheme outperforms the other three techniques. For Barbara image results are good for most of the payloads, but at payloads of higher embedding rate than 0.70 bpp the technique performs better than only one of the three competitive methods. This is because of increase in magnitude of prediction error for Barbara image on which MED predictor and interpolation seem to perform better on some prediction errors than Hybrid Predictor. The performance at specific points can be leveraged for specific types of images using adaptive embedding

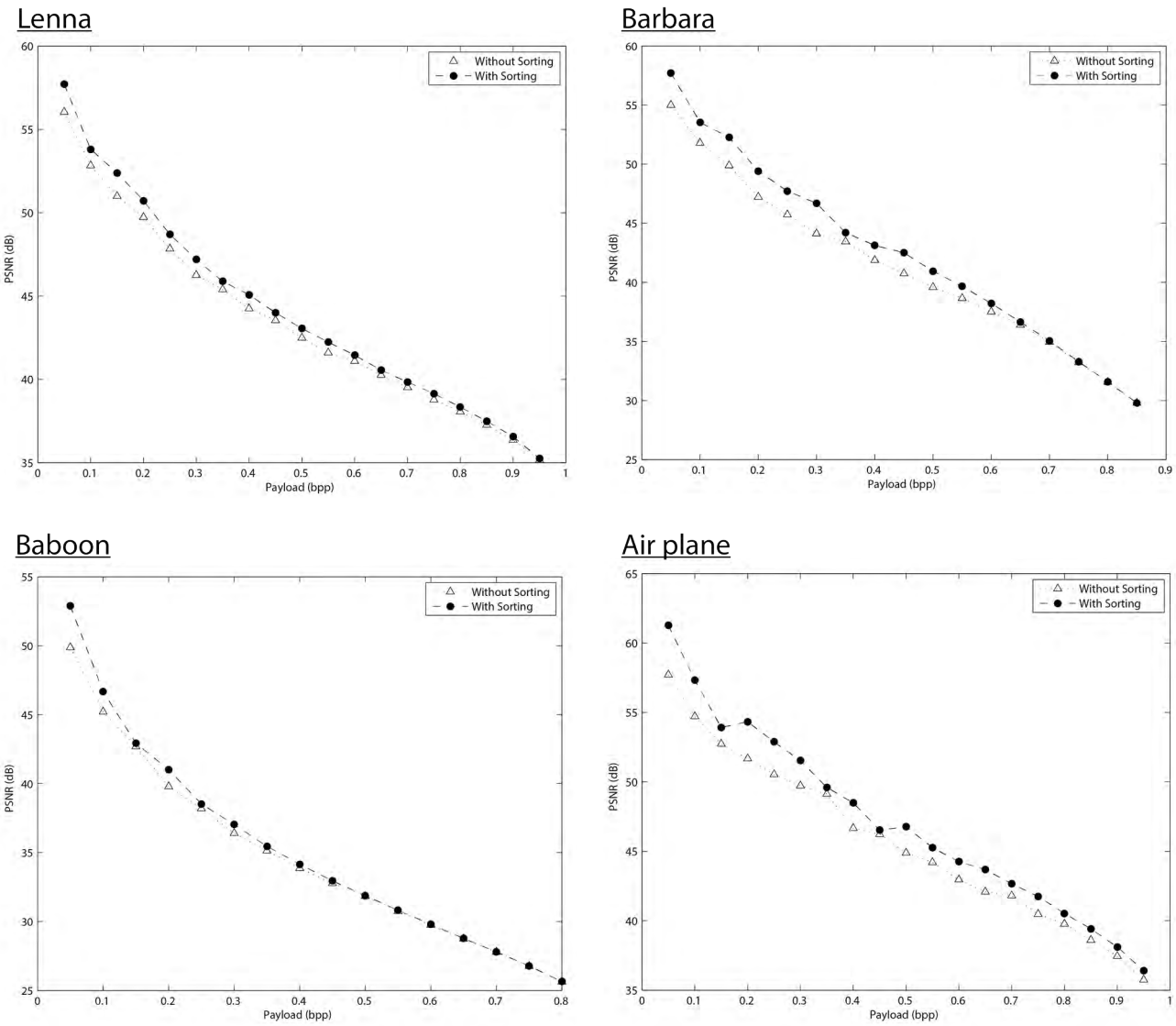

FIGURE 9. Impact of prediction error sorting on results. Prediction error sorting helps achieve lower image distortion for smaller payloads.

parameter tuning, but as we aim to propose a generalized embedding scheme so such parameter tuning is left as a future work for domain specific tuning.

The results for watermarked image distortion seems promising as we get higher PSNR for same payload sizes specially when the payload size is smaller. This bias for good performance at smaller payload is mainly caused because of sorting the prediction errors. Such a sorting allows us to embed the payload in those regions where a change will cause minimum distortion.

PSNR distortion measure statistics provides us a birds-eye view at the performance of proposed scheme. However, it is important to present the images produced during the process of watermarking to actually assess the distortion perceptible to human eyes. In figure 10 we present pictorial representations of the selected four sample images used for experimentation. There are five representations for every sample image that allows us a deeper insight into performance of proposed scheme:

- Images tagged with name "Original" (i.e. I-a Lenna Original , II-a Barbara Original, III-a Baboon Original and IV-a Plane Original) represents the original image before watermarking is done.
- Images tagged with name "Watermarked", represents the images watermarked using proposed approach.
- Images tagged with name "Extracted", represents images recovered by decoding the watermarked image. Because the proposed scheme is reversible so the extracted image should be same as original image.
- Images tagged with name "Difference Original & Watermarked" represents the difference image between original image and watermarked image. This

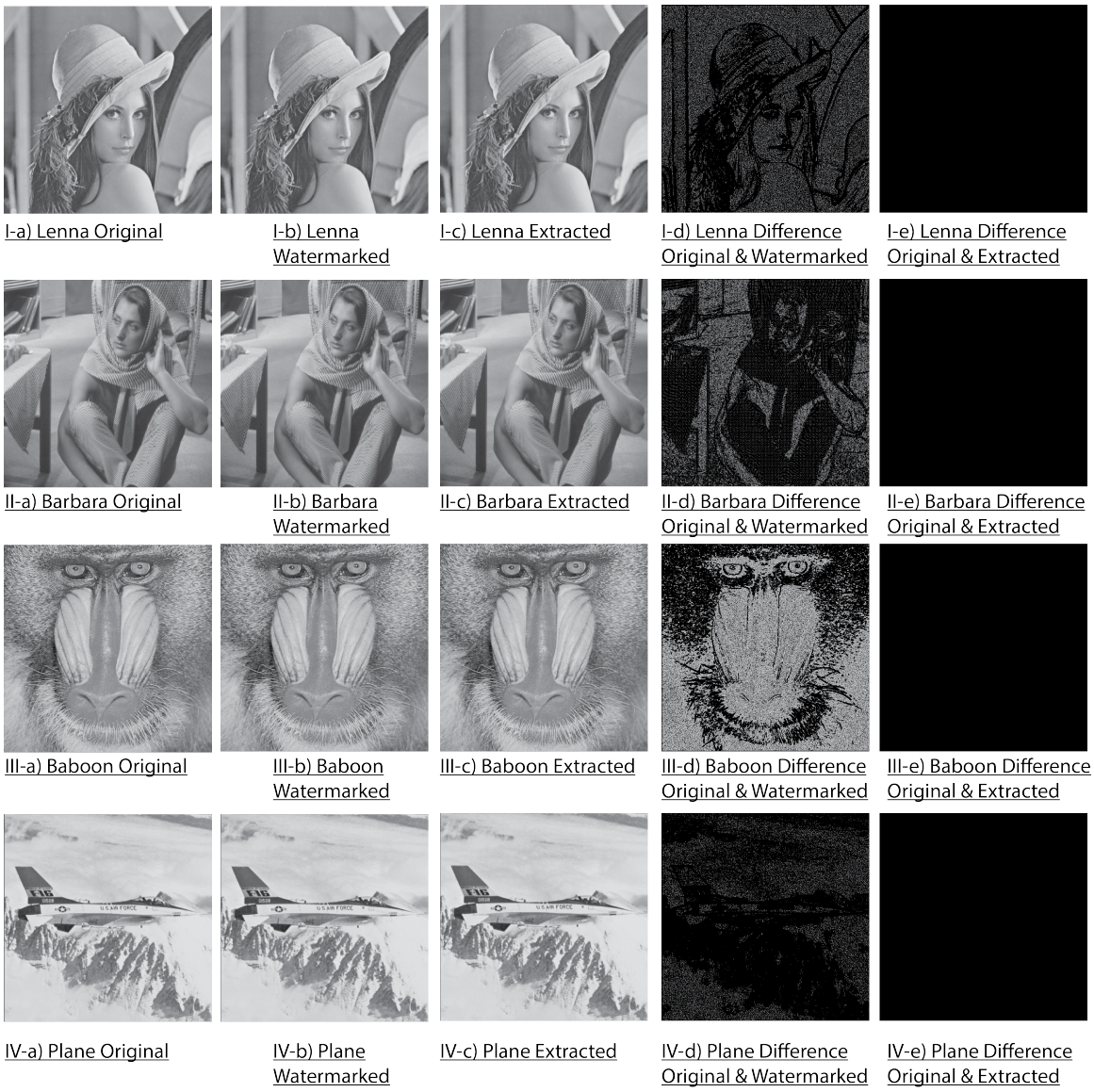

FIGURE 10. a. Original images before watermarking, b. Images watermarked using proposed approach, c. Extracted images decoded from watermarked images, d. Difference image between the watermarked and original image highlights the changed regions in the image, e. Difference image in original and extracted image shows no difference in the two.

representation gives an insight on modified regions where most of embedding data is stored. Flatter regions get most of the embedding data for the proposed scheme as can be observed from the given images. It is important to note that difference image is enhanced by scaling magnitude of difference image 20 times. This is because original magnitudes are not visible to the human eye on the image. Thus actual magnitude of the distortion should not be confused with intensity of the image. Rather the non–zero intensities should only be considered to represent as a representation highlighting regions that changed in the image.

- Images tagged with name ''Difference Original & Extracted'' represents difference image between original image and extracted image. Because proposed watermarking approach is ''reversible'' so the difference image between original image and extracted image is all zeros. This is evident by the black image consisting of all zeros in the difference image.

V. CONCLUSION

In this paper, a four phase reversible watermarking method and experiments performed on standard images show very promising results. Four phase representation provides larger prediction context for prediction error calculation while providing non-overlapping prediction contexts in every individual phase. Prediction error sorting allows increasing the probability of watermarking low magnitude prediction errors and thus causing less distortion. Adaptive embedding based upon local neighborhood variance to adaptively embed double data

in flatter regions for improving results for relatively larger embedding rates. Hybrid predictor predicts the values with more accuracy thus reducing the prediction error magnitudes resulting in lower image distortion for watermarked image. Importance of the technique is the flexibility and extendibility that it provides for future enhancements. The method can be converted into a framework which can be adjusted to tune and customize on domain specific problems.

REFERENCES

[1] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
[2] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.
[3] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," U.S. Patent 5 646 997, Jul. 8, 1997.
[4] D. M. Thodi and J. J. Rodríguez, "Prediction-error based reversible watermarking," in *Proc. Int. Conf. Image Process. (ICIP)*, Oct. 2004, pp. 1549–1552.
[5] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Process.*, vol. 92, no. 1, pp. 54–62, 2012.
[6] G. Xuan, Y. Q. Shi, J. Teng, X. Tong, and P. Chai, "Double-threshold reversible data hiding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2010, pp. 1129–1132.
[7] Y. Yang, X. Sun, H. Yang, C. T. Li, and R. Xiao, "A contrast-sensitive reversible visible image watermarking technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 656–667, May 2009.
[8] K.-L. Chung, Y.-H. Huang, P.-C. Chang, and H.-Y. M. Liao, "Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1643–1647, Nov. 2010.
[9] C.-H. Yang and M.-H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Process.*, vol. 4, no. 4, pp. 223–234, 2010.
[10] T.-Y. Liu and W.-H. Tsai, "Generic lossless visible watermarking—A new approach," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1224–1235, May 2010.
[11] M. Chaumont and W. Puech, "A high capacity reversible watermarking scheme," *Proc. SPIE*, vol. 7257, p. 72571H, Jan. 2009.
[12] W. Hong and T.-S. Chen, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting," *J. Syst. Softw.*, vol. 83, no. 12, pp. 2653–2663, Dec. 2010.
[13] Y.-C. Lin and T.-S. Li, "Reversible image data hiding using quad-tree segmentation and histogram shifting," *J. Multimedia*, vol. 6, no. 4, pp. 349–358, 2011.
[14] X.-R. Luo, C.-H. J. Lin, and T.-L. Yin, "Reversible data hiding based on two-level HDWT coefficient histograms," *Adv. Comput. Int. J.*, vol. 2, no. 1, pp. 1–16, Feb. 2011.
[15] H.-W. Yang, I.-E. Liao, and C.-C. Chen, "Reversible data hiding based on median difference histogram," *J. Inf. Sci. Eng.*, vol. 27, no. 2, pp. 577–593, 2011.
[16] C. Wang, X. Li, and B. Yang, "Efficient reversible image watermarking by using dynamical prediction-error expansion," in *Proc. Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 3673–3676.
[17] R. A. Farrugia, "A reversible visible watermarking scheme for compressed images," in *Proc. 15th IEEE Medit. Electrotech. Conf. (MELECON)*, Apr. 2010, pp. 212–217.
[18] C.-T. Li, "Reversible watermarking scheme with image-independent embedding capacity," *IEEE Proc.-Vis. Image Signal Process.*, vol. 152, no. 6, pp. 779–786, Dec. 2005.
[19] S.-C. Huang and M.-S. Lin, "A high-capacity reversible data-hiding scheme for medical images," *J. Med. Biol. Eng.*, vol. 30, no. 5, pp. 289–296, 2010.
[20] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 567–570, Jun. 2010.
[21] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Process.*, vol. 93, no. 1, pp. 198–205, 2013.
[22] H.-C. Huang, W.-C. Fang, and I.-T. Tsai, "Reversible data hiding using histogram-based difference expansion," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2009, pp. 1661–1664.
[23] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.
[24] K.-S. Kim, M.-J. Lee, H.-Y. Lee, and H.-K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognit.*, vol. 42, no. 11, pp. 3083–3096, Nov. 2009.
[25] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 906–910, Jun. 2009.
[26] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H.-G. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 456–465, Sep. 2008.
[27] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
[28] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.

MUHAMMAD ISHTIAQ received the B.S. degree from the Virtual University of Pakistan in 2007, and the M.S. degree from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2009, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His research interests include multimedia security and forensics.

WAQAR ALI received the B.S. degree from the COMSATS Institute of Information Technology, and the M.S. degree from the National University of Computer and Emerging Sciences, Islamabad, Pakistan. He is currently a Principal Software Engineer with GShare, Pakistan.

WASEEM SHAHZAD received the B.S. degree from UAAR, Rawalpindi, in 2004, and the M.S. and Ph.D. degrees from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2008 and 2010, respectively. He is currently a Professor and the Head of the Department with the National University of Computer and Emerging Sciences. He has research interests in the fields of data mining, computational intelligence, machine learning, and theory of computation. He has several research publications in these areas.

MUHAMMAD ARFAN JAFFAR received the B.Sc. degree (Hons.) from Bahauddin Zakariya University, Multan, Pakistan, in 2000, the M.Sc. degree in computer science from Quaid-i-Azam University, Islamabad, Pakistan, in 2003, and the M.S. degree in computer science and the Ph.D. degree from the National University of Computer and Emerging Sciences, Islamabad, in 2006 and 2009, respectively. He was a Research Professor with the Signal and Image Processing Laboratory, School of Mechatronics, Gwangju Institute of Science and Technology. He is currently an Assistant Professor with Al Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia. His research interests include image processing and computational intelligence.

YUNYOUNG NAM received the B.S., M.S., and Ph.D. degrees in computer engineering from Ajou University, South Korea, in 2001, 2003, and 2007, respectively. He was a Senior Researcher with the Center of Excellence in Ubiquitous System from 2007 to 2010. He was a Research Professor with Ajou University from 2010 to 2011. From 2009 to 2013, he was a Post-Doctoral Researcher with the Center of Excellence for Wireless and Information Technology, Stony Brook University, Stony Brook, NY, USA. He was a Post-Doctoral Fellow with the Worcester Polytechnic Institute, Worcester, MA, USA, from 2013 to 2014. He has been the Director of the ICT Convergence Rehabilitation Engineering Research Center with Soonchunhyang University, since 2017, where he is currently an Assistant Professor with the Department of Computer Science and Engineering. His research interests include multimedia database, ubiquitous computing, image processing, pattern recognition, context-awareness, conflict resolution, wearable computing, intelligent video surveillance, cloud computing, biomedical signal processing, rehabilitation, and healthcare system.