

Pixel-QUAD based Reversible Data Hiding for High Embedding Capacity

Rushikesh Borse

PhD Research Scholar, Vision and Image Processing lab
Department of Electrical Engineering, Indian Institute of Technology Bombay
Mumbai-400076, India

ABSTRACT

Embedding capacity and Perceptibility are prime issues in relation to data hiding techniques. There had been techniques for hiding data in reversible fashion. A new term 'Pixel-QUAD' is coined in this paper, instead of commonly used 'Pixel-Pair' for embedding the watermark data. This paper presents a high embedding capacity technique for reversible data hiding based on a Pixel-Quad approach. The proposed methodology for the technique is quite general and is applicable to any reversible watermarking scheme; Demonstration is done for a class of reversible watermarking schemes which operate on a disjoint group of pixels. An attempt is also made to propose algorithm to estimate the embedding capacity iteratively for a multi-pass scheme.

Keywords:

Data hiding, Reversible watermarking, Embedding capacity, Pixel-Quad, Capacity estimation.

1. INTRODUCTION

Reversible Watermarking [4] is a technique used to preserve the copy-right of digital data (image, audio and video), while at the same time it ensures exact recoverability of the watermark as well as the cover image. This finds application in few cases wherein along with the watermark, the cover image is also to be retrieved after watermark retrieval. There had been many techniques of reversible data hiding proposed in literature.

Various reversible data embedding techniques could be categorized under the heads of: histogram bin shifting [15], lossless data compression [12], expansion and mapping based techniques [9],[1],[6],[5] and prediction based techniques [11],[14]. Roverto *et al.* [13] and Feng *et al.* [7] wrote decent survey papers on different reversible watermarking techniques. Of all these techniques, difference expansion based techniques are prominently used for their advantage of high embedding capacity and less computational cost. In this regard, the general trend is to hide watermark data (a bit of data) into pixel pair of the cover image and this has been proposed by Tian [9], wherein the basic wavelet lifting transform (based on difference expansion) embed a bit of watermark into the LSB of a pixel pair. The inverse transform assures recoverability of watermark bit as well as the pixel pair values. This idea of reversible data hiding was soon generalized by Alattar [1]. Since

then many researchers have worked on this concept of difference expansion (DE) and proposed various difference expansion based techniques.

1.1 Embedding capacity

Embedding capacity is an important issue for reversible watermarking as one can hide more data with fewer computations and with a reasonably good perceptual quality. It is therefore that, reversible watermarking is also called reversible data hiding technique. Loosely speaking, the embedding capacity of an image can be described as the size of the largest watermark which can be embedded into that image. Each watermarking technique has a maximum possible embedding capacity over a single pass, and hence often it is necessary to go for multiple passes to embed a much larger watermark into the given image. Recall that multipass embedding involves at every stage, successively embedding the watermark bits into the already watermarked image from the previous stage. Several researchers had proposed various watermarking schemes to achieve higher embedding capacities. But obvious, every scheme has its own computational cost and memory usage which decides its application and usability. Tian [9] was the first to propose a difference expansion based watermarking scheme which work on a pair of pixel to embed a bit of data. Later on this concept was extended by Alattar [1] to triplets and quadruplets. Both this techniques require location map to achieve reversibility. This location map further reduces the embedding capacity. Later Coltuc *et al.* [5],[6] proposed a difference expansion based reversible data hiding technique which does not require a location map thus making it simple to implement with advantage of modest embedding capacity. The technique proposed by Coltuc *et al.* [6] is discussed in detail in section 2.1.

1.2 Contribution of the paper

The intent of this paper is not to propose a new watermarking scheme rather describe an approach for achieving higher embedding capacity for the existing watermarking scheme. This is achieved by using Pixel-Quad approach instead of commonly used Pixel-Pair.

Secondly: After proposing the Pixel-Quad approach of data embedding, an effort is also made to propose algorithm to estimate the embedding capacity for any given watermarking scheme. Any wa-

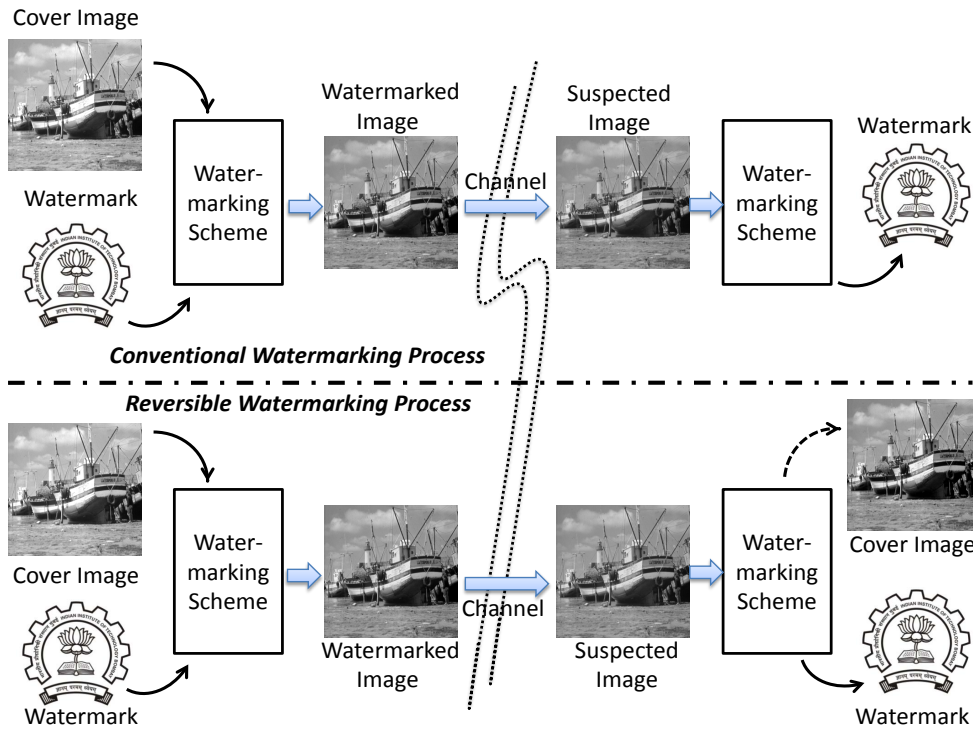


Fig. 1: Conventional and Reversible Watermarking scheme

watermarking application would require an estimation of the number of passes of watermarking possible as well as an analysis of the feasibility of inserting a watermark of specific length into a given image. For this purpose it is necessary to calculate the embedding capacity beforehand. In practical settings, it may be necessary to find such estimates repeatedly for different configurations of the watermark and the cover image. Hence it may not be feasible to actually embed the watermark in a given image, and to check if the watermark and the cover image are compatible for watermarking. The paper addresses these two issues.

Although the mathematics discussed in quite general in its sense and is applicable to any watermarking scheme which work on disjoint pixel pair, experimentation of these algorithms are done on RCM scheme (reversible contrast mapping) as proposed by Coltuc *et al.* [6]. Next Section describes the Methodology for data embedding for Quad. Section 3 discusses a very important issue i.e. estimating the embedding capacity for this Quad representation. Results are provided in Section 4 which justifies the concept. The paper concludes in Section 5. As far as this paper is concerned, the word watermarking, data hiding and data embedding means the same unless stated otherwise in this paper.

2. PROPOSED METHODOLOGY

As mentioned previously, reversible watermarking in general is a data hiding technique for exact recovery of both watermark and the cover image at receiver. This can be seen from Figure 1. As the name speaks 'reversible', one would be interested to hide more and more data into the cover image. Of various data hiding techniques [1],[9],[15],[12],[6],[10] difference expansion based techniques are popular because they can hide ample amount of data with minimal visual distortion. For obtaining higher embedding capacity, re-

searchers devised multi-pass embedding technique, wherein watermarking is done in consecutive passes to the cover image and then to the watermarked image. The author has addressed this subject in [3]. All this techniques are based on Pixel-Pair approach. Taking the motivation further, A Pixel-Quad based embedding scheme is proposed. A quad is composed of 4 pixels as can be seen in Figure 2a. This can be viewed as a combination of two pixel pairs and the watermarking (data embedding or data hiding) could be done vertically or horizontally as seen in Figure 2b and c. Thus a quad can embed a maximum of 2 bits of data (If both the pixel pairs satisfy the constraint condition).

2.1 Pixel-QUAD for data hiding

A Pixel-Quad can be seen as a combination of two pixel-pairs, thus the technique of data hiding for quad is closely associated with pair. The proposed framework is quite general and applicable to any watermarking schemes working on disjoint pixel pair. The proposed algorithm is evaluated on well known Reversible Contrast Mapping (RCM) based watermarking scheme [6],[5]. Let's recall the generalized form of the method as mentioned by Coltuc *et al.* [5]: For a pixel pair (x, y) , the forward mapping transforms them into another pair (x', y')

$$x' = (n + 1)x - ny, \quad y' = -nx + (n + 1)y \quad (1)$$

while the inverse transform $(x, y) = T^{-1}(x', y')$ can be obtained as:

$$x = \frac{(n + 1)x' + ny'}{2n + 1}, \quad y = \frac{nx' - (n + 1)y'}{2n + 1}. \quad (2)$$

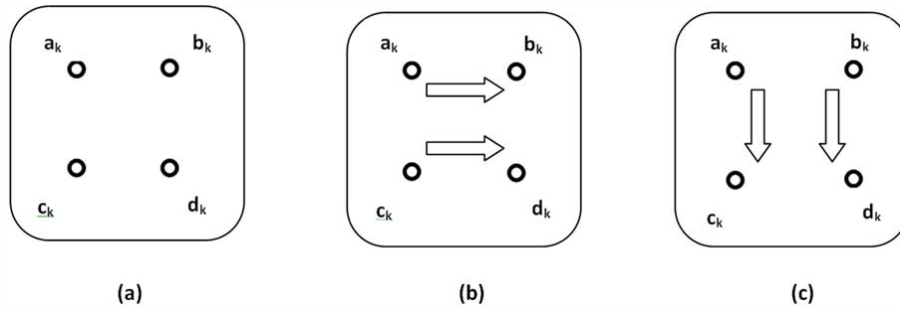


Fig. 2: Directions of data embedding in Pixel-Quad

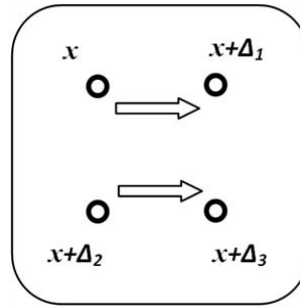


Fig. 3: A simple Quad for explaining difference expansion after forward transform.

Specifically dealing with the case $n = 1$, above equations reduces to:

$$x' = 2x - y, \quad y' = 2y - x \quad (3)$$

while the inverse transform is becomes:

$$x = \lceil \frac{2}{3}x' + \frac{1}{3}y' \rceil, \quad y = \lceil \frac{1}{3}x' - \frac{2}{3}y' \rceil \quad (4)$$

where $\lceil x \rceil$ is called as the ceil function.

Because of equation (2), the original pixel values can be recovered from the transformed one, even if their LSB's are lost or altered. (Except for the case where LSB's of both original pixel pairs are odd, where few adjustments are done). Taking the advantage of this condition, one can embed a bit of information in LSB's of transformed pairs.

To avoid ambiguity and to prevent overflow and underflow, the gray scale values of transformed pixel pairs are restricted within a sub-domain S_r defined as

$$S_r = \{(x', y') \mid x' \in [0, L), y' \in [0, L)\} \quad (5)$$

This is constraint equation, where L is the number of graylevels (In this paper, 8-bit gray scale image is considered; $L = 2^8 = 256$).

2.2 Difference Error over Multipass Embedding

As the name RCM dictates, there is contrast expansion after forward transform (equation (3)). Although this expansion is reversible (equation (4)), let's say this as difference error, which is represented as:

$$D_e = |x' - y'| = 3|x - y| = 3\Delta \quad (6)$$

This increase in contrast leads to the reduction of PSNR in the watermarked image. Let's extend this to Quad. Consider the quad as shown in Figure 3, where horizontal pixel-pairing is considered. For the sake of simplicity, let's say: $a_k = x$, $b_k = x + \Delta_1$, $c_k = x + \Delta_2$ and $d_k = x + \Delta_3$. Applying RCM transform equation (3) for the first pair (a_k, b_k) and second pair (c_k, d_k) would result into:

$$a'_k = 2x - (x + \Delta_1), \quad b'_k = 2(x + \Delta_1) - x \quad (7)$$

&

$$c'_k = (2x + 2\Delta_2) - (x + \Delta_3), \quad d'_k = (2(x + \Delta_3) - (x + \Delta_2)) \quad (8)$$

The corresponding error for both the transformed pairs with respect to original pairs would be:

$$D_{e1} = |a'_k - b'_k| = 3|a_k - b_k| = 3\Delta_1 \quad (9)$$

&

$$D_{e2} = |c'_k - d'_k| = 3|c_k - d_k| = 3(\Delta_2 - \Delta_3) \quad (10)$$

The error so obtained in these equations is in accordance with equation 6. So the effective error after first pass is 3Δ . Considering the same pixel pairs inside the quad (this means embedding in same horizontal direction), the difference error after second pass of data embedding would be 9Δ and this goes on increasing in further pass. This results in reduction of embedding capacity in further passes as most of the pixels pairs may not be eligible for data embedding after 2-3 passes. Hence the embedding capacity get exhausted earlier. A theorem is proposed in this regard in [3]. Thus the over error after n passes of watermarking, would be $(3\Delta)^n$.

It is thus suggested to embed the data in a vertical fashion in second pass (as first pass was in horizontal fashion). This would also

result a increase in the difference error but the rate of increase of error would be relatively lower. After two passes it would be less than 6Δ . A generalized error equation for alternate passes of embedding is derived in [2]. Thus the pixel quad will survive more passes and embed more data, resulting into a high data embedding scheme at same PSNR. It is thus learned that data embedding in pixel quad be done in alternate direction for multiple passes and not in a same direction. Results for Pixel-Quad embedding are shown in Section 4, where this proposal is experimentally verified that final embedding capacity obtained for alternate direction of watermarking results in higher embedding capacity.

2.3 How to decide the direction of first pass?

Coming to the understanding that watermarking is done in alternate direction of pairing of pixels, but how to choose the direction at the first pass?. It may be vertical or horizontal. A simple mathematics of finding the variation of data set along vertical and horizontal direction would be sufficient.

$$Variation_{horizontal} = \sum_{i=1}^N \sum_{j=2n}^M (X_{i,j} - X_{i,j+1})^2 \quad (11)$$

$$Variation_{vertical} = \sum_{i=2n}^N \sum_{j=1}^M (X_{i,j} - X_{i+1,j})^2 \quad (12)$$

Where X is the input image (data set).

The lowest of the two reflects the variation is less in that direction, thus suitable for data embedding for first pass. After first stage, the direction for embedding the data for consecutive passes is alternate and it is observed from practical implementation that the embedding capacity is improved. In fact in some cases, there has been sufficient space to allow one more pass of embedding, thus resulting into a higher embedding capacity. This is discussed in results section and the values are reported in Table 1.

3. ESTIMATION OF EMBEDDING CAPACITY FOR PIXEL-QUAD

Having proposed a Pixel-Quad based methodology for data embedding, an attempt is made to provide an estimate for the embedding capacity. Estimate in general sense means that one would be able to know the amount of embedding capacity that is available for a given cover image in advance. Thus one could choose the cover image or reject it based on the available watermark. All this is obtained without having to do actual watermarking.

3.1 Algorithms Involved

The estimation algorithm is based on two prominent algorithms: 1. Co-occurrence matrix based algorithm and 2. Tree based algorithm. Both these algorithms are elaborately discussed in [2], lets describe them in short.

DEFINITION 1. (Pair-wise co-occurrence Matrix). The pair-wise co-occurrence matrix C of size $L \times L$, is similar to the conventional co-occurrence matrix [8], and is defined as the population (distribution) of co-occurring pixel pairs in an Image. In this context, it represents a count of the number of times a pixel

pair occurs in the image. Thus given the disjoint pairs of pixels $\Xi = \{\xi_1, \xi_2, \dots, \xi_N\}$. It can be mathematically described as:

$$C(\xi) = \sum_{j=1}^N I(\xi = \xi_j), \text{ where } I(.) \text{ is the indicator function.} \quad (13)$$

Co-occurrence based method is an iterative procedure similar to any watermarking scheme. It is however computationally much more efficient than any stage-wise watermarking scheme because of less computations and sparse matrix representation. Thus gives an faster estimate.

DEFINITION 2. (Pixel-pair tree): The pixel-pair tree for a pair ξ is defined as a tree which starts with the pixel pair ξ and traces a specific path based on the subsequent embedded bits as this pixel pair evolves and represents all feasible paths of watermarking for a given number of passes.

As mentioned earlier, a quad is nothing but a combination of two pixel pairs, the algorithm and mathematics for quad is just an extension for that of pair. A pixel pair tree for a pixel pair (50, 55) using the RCM based watermarking scheme of Coltuc [6], is shown in Figure 5. This tree concisely represents the entire life cycle of a particular pixel pair upto P passes. It can be seen that the entire embedding exhausts after 3 passes.

For any pixel pair there could be three possibilities of embedding: (i) either a bit '1' is embedded, (ii) or a bit '0' is embedded, (iii) or nothing is embedded, if the pixel pair doesn't satisfy the constraint equation 5. Let's say this is ϕ . Extending this to a Pixel-Quad, where one could embed a maximum of two bits, there are 9 possibilities. This is explained in pictorial form in Figure 4.

3.2 Capacity Estimation

The formulations for pixel-pair are derived in [2]. General notations, terms and basic math continues from there. Lets extend this to Pixel-Quad. In general, let p_s denote the path specific probability of the path s . It represents the probability that the pixel pair will evolve through the specific path s . Then clearly, one can write

$p_s = \prod_{k=0}^{P-1} p_{s[k]}$, where $p_{s[k]}$ denotes the probability of transition from $s[k]$ to $s[k+1]$ given whether $s[k]$ is embeddable. Thus

$$p_{s[k]} = \begin{cases} p & , \text{ if } i_{s[k]} = 1 \ \& \ s[k] \in \mathbb{D}_T \\ 1 - p & , \text{ if } i_{s[k]} = 0 \ \& \ s[k] \in \mathbb{D}_T \\ 1 & , \text{ if } s[k] \notin \mathbb{D}_T \end{cases} \quad (14)$$

The equation above represents the probability for going to further pass, where \mathbb{D}_T represents the domain where the transformed pixel satisfy the constraint equation 5.

Let $\eta(s, \mathcal{B})$ represent the size of the bitstream obtained through a specific path configuration $s = \{s_1, s_2, \dots, s_N\}$. Further let $\eta(\xi, s, \mathcal{B})$ denote the total number of bits contributed by a specific pixel pair ξ through the path s . In other words $\eta(\xi, s, \mathcal{B}) = \sum_{k=0}^{P-1} \eta(b_{s[k]})$ and $\eta(s, \mathcal{B}) = \sum_{j=1}^N \eta(\xi_j, s_j, \mathcal{B})$. The estimate of the length of the bitstream \mathcal{B} can then be given as $\eta(\mathcal{B}) = \mathbf{E}_{s \in \mathbb{S}}(\eta(s, \mathcal{B}))$, where $\mathbf{E}_{s \in \mathbb{S}}(\eta(s, \mathcal{B}))$ represents the total expected size of \mathcal{B} by considering every possible path configuration $s_1 \in \mathbf{S}_{\xi_1}, s_2 \in \mathbf{S}_{\xi_2}, \dots, s_N \in \mathbf{S}_{\xi_N}$, which one can also write as $s \in \mathbb{S}$.

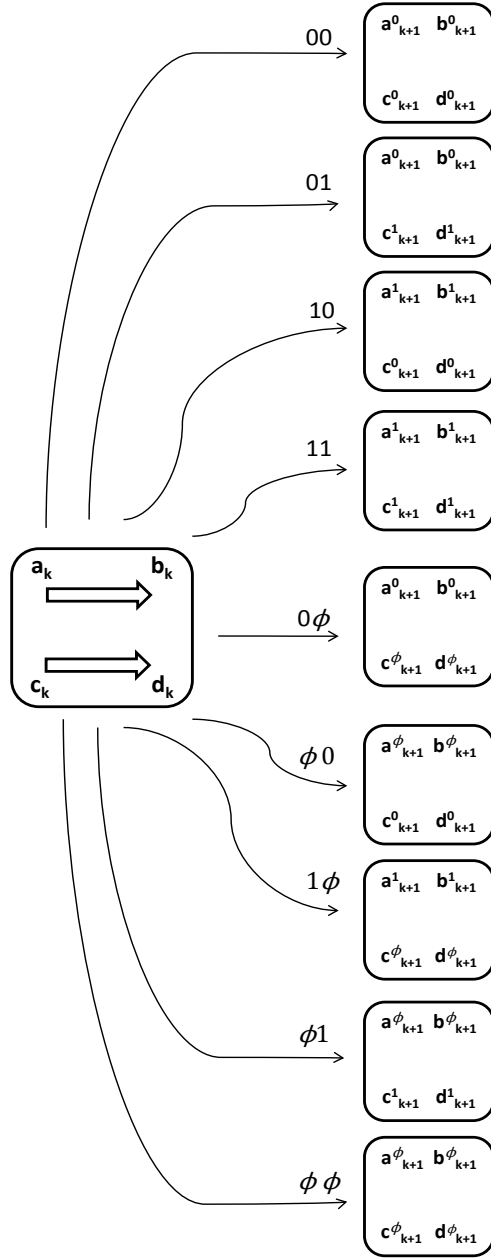


Fig. 4: An Illustration of possible combination of embedding bits for a Pixel-Quad.

The expressions for $\mathbf{E}_P^\eta(\xi, \mathcal{B})$ (for $P > 1$) is as:

$$\mathbf{E}_P^\eta(\xi, \mathcal{B}) = \begin{cases} \eta(b_\xi) + p \mathbf{E}_{P-1}^\eta(\xi^1, \mathcal{B}) \\ \quad + (1-p) \mathbf{E}_{P-1}^\eta(\xi^0, \mathcal{B}) & \text{, if } \xi \in \mathbb{D}_I \\ \eta(b_\xi) + \mathbf{E}_{P-1}^\eta(\xi^\phi, \mathcal{B}) & \text{, if } \xi \notin \mathbb{D}_I \end{cases} \quad (15)$$

with the base case: $\mathbf{E}_1^\eta(\xi, \mathcal{B}) = \eta(b_\xi)$. Referring to Figure 4, it is understood that a quad can embed maximum up to 2 bits, lets

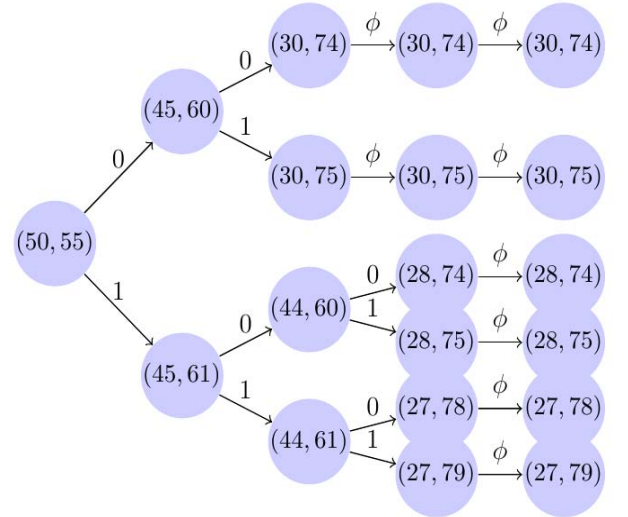


Fig. 5: An example of pixel-pair tree for (50,55)

extend equation 14 for the case of Pixel-Quad as:

$$p_{s[k]} = \begin{cases} p^2 & \text{, if } i_{s[k]} = 1, 1 \text{ \& } s[k] \in \mathbb{D}_I \\ p(1-p) & \text{, if } i_{s[k]} = 1, 0 \text{ \& } s[k] \in \mathbb{D}_I \\ (1-p)p & \text{, if } i_{s[k]} = 0, 1 \text{ \& } s[k] \in \mathbb{D}_I \\ (1-p)^2 & \text{, if } i_{s[k]} = 0, 0 \text{ \& } s[k] \in \mathbb{D}_I \\ p & \text{, if } i_{s[k]} = 1, \phi \text{ or } \phi, 1 \text{ \& } s[k] \in \mathbb{D}_I \\ (1-p) & \text{, if } i_{s[k]} = 0, \phi \text{ or } \phi, 0 \text{ \& } s[k] \in \mathbb{D}_I \\ 1 & \text{, if } i_{s[k]} = \phi, \phi \text{ \& } s[k] \notin \mathbb{D}_I \end{cases} \quad (16)$$

In similar regards, equation (15) can be extended to quad as:

$$\mathbf{E}_P^\eta(\xi, \mathcal{B}) = \begin{cases} \eta(b_\xi) + (1-p)^2 \mathbf{E}_{P-1}^\eta(\xi^{00}, \mathcal{B}) \\ \quad + (1-p)p \{ \mathbf{E}_{P-1}^\eta(\xi^{01}, \mathcal{B}) \\ \quad + \mathbf{E}_{P-1}^\eta(\xi^{10}, \mathcal{B}) \} \\ \quad + p^2 \mathbf{E}_{P-1}^\eta(\xi^{11}, \mathcal{B}) & \text{, if 2 bits are embedded \& } \xi \in \mathbb{D}_I \\ \eta(b_\xi) + p \{ \mathbf{E}_{P-1}^\eta(\xi^{1\phi}, \mathcal{B}) \\ \quad + \mathbf{E}_{P-1}^\eta(\xi^{\phi 1}, \mathcal{B}) \} \\ \quad + (1-p) \{ \mathbf{E}_{P-1}^\eta(\xi^{\phi 0}, \mathcal{B}) \\ \quad + \mathbf{E}_{P-1}^\eta(\xi^{0\phi}, \mathcal{B}) \} & \text{, if 1 bit is embedded \& } \xi \in \mathbb{D}_I \\ \eta(b_\xi) + \mathbf{E}_{P-1}^\eta(\xi^{\phi\phi}, \mathcal{B}) & \text{, if no bits are embedded \& } \xi \notin \mathbb{D}_I \end{cases} \quad (17)$$

with the base case: $\mathbf{E}_1^\eta(\xi, \mathcal{B}) = \eta(b_\xi)$.

It is clear from equation (17) that $\mathbf{E}_P^\eta(\xi, \mathcal{B})$ or the total expected number of bits for a given pixel pair is a polynomial equation in p . Hence in order to make the offline stage independent of p , let's compute the polynomial coefficients of $\mathbf{E}_P^\eta(\xi, \mathcal{B})$ for every pixel pair and store them. Thus one can directly use these stored coef-

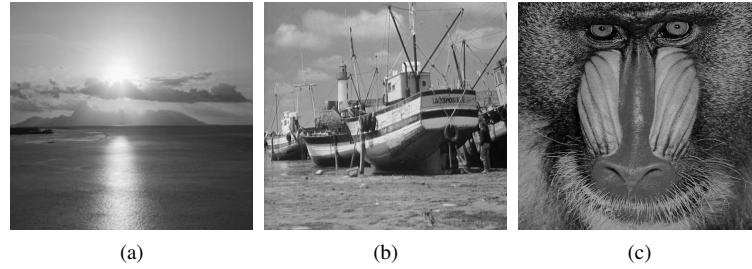


Fig. 6: A few sample images of varying textural contents for watermark embedding purposes.

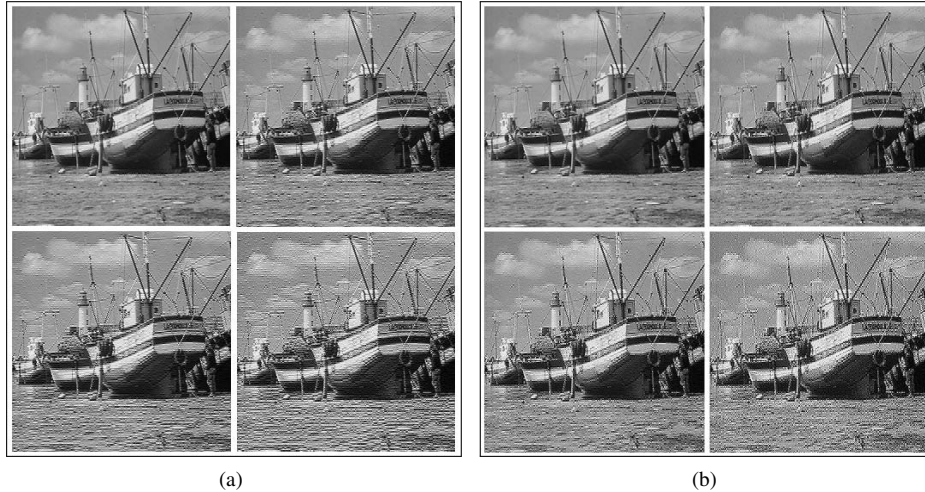


Fig. 7: Watermarked copies of *Boat* image after multiple passes of embedding. These are results of watermarking in same direction (fig a) and watermarking in alternate direction (fig b). These images are of (i) First pass, (ii) Second pass, (iii) Third pass and (iv) Fourth pass.

ficients to compute $\mathbf{E}_p^\eta(\xi, \mathcal{B})$ for any given p without having to recompute them every time.

Thus there are two stages of computation, i.e the online stage and offline. The offline stage is image independent where one compute the coefficients of the polynomials, as described in equation (17). These can be iteratively computed for various values of P in linear time. The online stage consists of just a single iteration to use the values of $\mathbf{E}_p^\eta(\xi, \mathcal{B})$ which has to be estimated.

Hence the total embedding capacity can be figured out as:

$$\eta(\mathcal{E}) = \eta(\mathcal{I}) - \eta(\mathcal{A}) \quad (18)$$

Where \mathcal{I} is the total embeddable bit stream and \mathcal{A} is the auxiliary data bits that comprises of LSB's of pixel pairs that does not satisfy constraint equation (eqn 5) are thus not eligible candidate for embedding. These are required to recover the original pixel values at the receiver side.

This sub-section discussed method of estimating embedding capacity based on tree based algorithm. Another method of estimating embedding capacity is based on co-occurrence matrix based. This is elaborately discussed in [2]. A competitive analysis of both these algorithms is also provided thither.

4. RESULTS AND DISCUSSION

The concept discussed in Section 2 and Section 3 was implemented on many images. For brevity, the results are presented for three commonly known images *Sunset.jpg*, *Boat.png* and *Baboon.gif*, all of size 256x256, as shown in Figure 6. Figure 7 shows results of watermarking *Boat* image for four times. As Proposed in Section 2, multiple watermarking be done in alternate direction in consecutive passes rather than in same direction. The difference in both the results can be visually compared. Results of consecutive passes are shown in Figure 7. Part-a (left side) of this figure shows results of watermarking in same direction, while Part-b (right side) shows the results in alternate direction. It can be easily appreciated that the results in Part-b outweigh the results in Part-a, especially figure (iii) and (iv) are quite better in Part-b as compared to those in Part-a. This not only preserves the quality of image in terms of PSNR but also provides more space for data embedding in terms of embedding capacity. It can be seen from results in Table 1 that embedding in alternate direction always gives more space for embedding. In fact for all the images, it even gives one extra pass of embedding. In *Sunset* image, there is embedding space available even in 6th pass (for alternate fashion of embedding), while the capacity gets exhausted if watermarking is done in same direction. The same is true for *Boat* and *Baboon* images too. A ‘-’ in the table 1 indicates

Table 1. : Comparison of stage wise embedding capacity for data hiding in different direction

Passes of embedding	Sunset.jpg (Capacity in bpp)		Boat.png (Capacity in bpp)		Baboon.gif (Capacity in bpp)	
	Same direction	Alternate direction	Same direction	Alternate direction	Same direction	Alternate direction
I	0.4998	0.4998	0.4842	0.4842	0.4471	0.4472
II	0.4871	0.4933	0.4033	0.4456	0.2580	0.3328
III	0.4458	0.4693	0.2821	0.3552	0.0799	0.1750
IV	0.3421	0.3840	0.1035	0.2453	–	0.0480
V	0.1825	0.2433	–	0.0518	–	–
VI	–	0.0608	–	–	–	–
Total Emb. Capacity	1.9574	2.1506	1.2732	1.5821	0.7852	1.0051

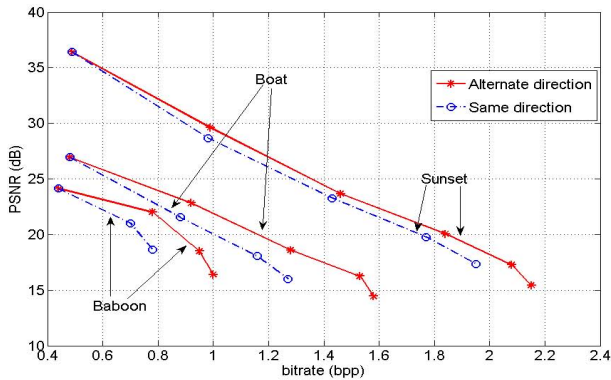


Fig. 8: Cumulative PSNR-bpp curve for different images.

that capacity is exhausted at that location and there is no possibility of data embedding any more (as most of the pixel-pairs does not satisfy constraint equation 5 and thus the available embedding capacity is been eaten up to store LSBs of these pixel-pairs and this contribute to auxiliary data bits A).

In general, the over all embedding capacity after all passes is always higher for data embedding when done in alternate direction. It can also be seen that although the stage wise embedding capacity reduces in every further pass of watermarking, the over all embedding capacity always rises as one goes for further passes of embedding. A cumulative plot for varying textured images (*Baboon*-highly textured image, *Sunset*-low textural image) is plotted in Figure 8. The embedding capacity and PSNR for alternate direction watermarking is always higher as compared to when done in same direction.

As far as estimation of embedding capacity is concerned, it is done in [2] for a pair of pixels. This is based on a Look Up Table(LUT) concept. The idea is to create a memory table for every pair and its all possible embedding options (tree) for many further passes. This is done for all possible pixel pairs. i.e from (0, 0) to (255, 255). This would be around 2^{256} by 2^{256} combinations (considering an eight bit image). This all is stored in a LUT. This is computed off-line, and one has to just get pixel-pair online for which he can get the embedding capacity as per the LUT. Now this same idea is to be extended to Pixel-Quad, but the LUT creation may suffer from the limitation of computational resources both in terms of time and memory, as it would require 4^{256} by 4^{256} combinations. All this is possible with a high end computationally efficient machine. The mathematics for quad estimation as derived in Section 4 is then applicable as it is for estimation of embedding capacity.

5. CONCLUSION

This paper is focused around providing a Pixel-Quad based solution for achieving high embedding capacity in any reversible watermarking scheme. It is proved from the experimental results that alternate direction of watermarking in consecutive passes would always result in a higher embedding capacity. Besides this, an effort is made to derive mathematics related to embedding capacity estimation of Pixel-Quad. This estimate is based on the tree based algorithm.

6. ACKNOWLEDGEMENT

The Author is very thankful to Rishabh Iyer of University of Washington, Seattle, USA for helping in the initial understanding of quad based approach.

7. REFERENCES

- [1] A.M.Alattar. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Processing*, 13(8):1147–1156, August 2004.
- [2] Rushikesh Borse. *Embedding Capacity Estimation for Reversible Watermarking Schemes and Data Security in H.264 Video*. PhD thesis, Indian Insitute of Technology Bombay, Mumbai, India, 2013.
- [3] Rushikesh Borse and Subhasis Chaudhuri. Computation of embedding capacity in reversible watermarking schemes. *Proceedings of the ACM's ICVGIP-10, Chennai, India*, 2010.
- [4] Ingemar Cox. *Digital Watermarking and Steganography*. Morgan Kaufmann, Second Edition, New York, 2008.
- [5] D.Coltuc and J-M Chassery. High capacity reversible watermarking. *Proceedings of the IEEE International Conference on Image Procesing ICIP*, pages 2565–2568, 2006.
- [6] D.Coltuc and J-M Chassery. Very fast watermarking by reversible contrast mapping. *IEEE Signal processing Letters*, 14(4):255–258, 2007.
- [7] J.B. Feng, I.C. Lin, C.S. Tsai, and Y.P. Chu. Reversible watermarking:current status and key issues. *International Journal of Network Security*, 12(3):161–170, 2006.
- [8] R.M. Haralick, K. Shanmugam, and I.H. Dinstein. Textural features for image classification. *IEEE Transactions on systems, man and cybernetics*, 3(6):610–621, 1973.
- [9] J.Tian. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Systems for Video Technology*, 13(8):890–896, August 2003.

- [10] C.-T. Li. Reversible watermarking scheme with image-independent embedding capacity. *IEE Trans on Vision, Image and Signal Processing*, 152(6)(779-786), 2005.
- [11] L.Kamastra and H.Heijmans. Reversible data embedding into images using wavelet techniques and sorting. *IEEE Trans. Image Processing*, 14(12):2082–2090, December 2005.
- [12] M.U.Celik and G.Sharma. Lossless generalized-lsb data embedding. *IEEE Transactions on Image Processing*, 14(2):253–266, 2005.
- [13] C. Roberto, F. Francesco, and B. Rudy. Reversible watermarking techniques: An overview and a classification. *EURASIP Journal on Information Security*, 2010.
- [14] Vasilij Sachnev. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Systems for Video Technology*, 19(7):989–999, July 2009.
- [15] C. D. Vleeschouwer, J. E. Delaigle, and B. Macq. Circular interpretation of histogram for reversible watermarking. *in Proceedings of the IEEE 4th Workshop on Multimedia Signal Processing*, pages 345–350, Oct 2001.