Computation of Embedding Capacity in Reversible Watermarking Schemes

Rushikesh Borse* Subhasis Chaudhuri
Vision and Image Processing Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
Powai, India
{rpborse, sc} @ ee.iitb.ac.in

ABSTRACT

This paper deals about embedding capacity computation for reversible watermarking schemes. The paper proposes a unique way of computing embedding capacity directly from the data set without actually embedding the watermark in the image. This computation is done based on the statistical parameters of the data set. We also demonstrate how to compute the capacity under distortion constraints. We also show how to enhance the capacity by using a multi-pass embedding scheme without substantially affecting the PSNR.

Keywords

Reversible watermarking, Embedding capacity, Multipass embedding, Control distortion

1. INTRODUCTION

As against robust watermarking [2] which talks about attacks and techniques to resist this attacks, we have another scheme called reversible watermarking. Here an exact recoverability of the watermark is of prime concern. Reversible watermarking 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. The concept of hiding a bit of information in the LSB plane is quite known since long [2] as it maintains the perceptual quality of image even after watermarking. As far as inserting a bit in a pair of pixels is concerned, it was initially proposed by Tian [8] and then later on generalized by Alattar [1]. There had been many papers in this regard [5, 10, 11], but most of them someway or other have used a data compression algorithm or they suffer from the low embedding capacity issue [8, 11].

Tian [8] initally proposed the difference expansion scheme which applies the 1-D Haar wavelet transform to the image and embeds the watermark into high-frequency coefficients by difference expansion. Alattar [1] extended Tian's scheme and applied the difference expansion (DE) of a generalized integer transform to pixels for reversible embedding. Kamstra and Heijmans [10] improved Tian's method by sorting the least-significant bits (LSB's) of pixel pairs to improve the coding efficiency of the lossless compression. Thodi et al. [5] introduced histogram shifting technique to Tian's scheme. Coltuc et al. [4] used a reversible contrast mapping (RCM) to embed watermark.

Embeding capacity is an important issue for reversible watermarking as one can hide more data with less computations and with a reasonably good perceptual quality. It is therefore that, reversible watermarking is also called reversible data hiding technique. In this paper, we extend the work proposed by Coltuc et al. [4] and propose a new technique of calculating embedding capacity for an image using the pair-wise co-occurance matrix (see section 3 for defination) which is computationally very efficient. The pair-wise co-occurance matrix is computed for different displacements, and the one which gives the maximum embedding capacity is chosen. This concept is significant as one can know the embedding capacity of a given image in advance before actually embedding it, based on which the dataset (given cover image) may be selected or rejected as the case may be. The paper also talks about the PSNR-bpp relationship and how to select the optimum value of control distortion factor for a specified PSNR. Although Li et al. [9] talks about image independent embedding capacity which involves a Hamming code based algorithm, but it is restricted to a very low embedding capacity (under 0.02 bpp).

2. REVERSIBLE WATERMARKING

We discuss the basics of reversible watermarking in this section. The difference expansion technique as mentioned by Tian [8] embeds a bit (either 0 or 1) in the difference h. A combination of average l and difference h for a pair of pixels *(x,y)* is given as:

$$l = \lfloor (x+y)/2 \rfloor \quad (1)$$

$$h = (x-y) \quad (2)$$

where $\lfloor$ x $\rfloor$ is called as the floor function. Inverse trans-

*On Duty leave and under financial assistance from Sinhgad Academy of Engineering Pune, India

ICVGIP '10, December 12-15, 2010, Chennai, India

form yields the value of this pixels x and y as:

$$x = l + \lfloor (h+1)/2 \rfloor \tag{3}$$

$$y = l - h/2. \tag{4}$$

The difference h is expanded to create a new LSB to embed a watermark bit w, and the watermark difference h_w is:

$$h_w = 2h + w. \tag{5}$$

The following equation is to be satisfied for overflow and underflow conditions to be avioded:

$$|h_w| \in D(l) = [0, min(2(2^n - 1 - l), 2l + 1)] \tag{6}$$

where n indicates the bits used for representation of a pixel value. Tian's scheme can achieve high capacity, however, it is difficult to implement capacity control due to the need to embed a compressed location map along with the payload. This decreases the embedding capacity further more. This problem is generalized by Coltuc et al [3] for the general case, and dealt specifically in another paper based on reversible contrast mapping for a pair of pixels in [4]. The RCM method is briefly described here:

For a pixel pair (x, y), the forward mapping transforms them into another pair (x', y')

$$x' = (n+1)x - ny, \quad y' = -nx + (n+1)y \tag{7}$$

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

$$x = \frac{(n+1)x' + ny'}{2n+1}, \quad y = \frac{nx' - (n+1)y'}{2n+1}. \tag{8}$$

Specifically dealing with the case $n = 1$, above equations become [4]

$$x' = 2x - y, \quad y' = 2y - x \tag{9}$$

while the inverse transform is defined as:

$$x = \lceil \frac{2}{3}x' + \frac{1}{3}y' \rceil, \quad y' = \lceil \frac{1}{3}x' - \frac{2}{3}y' \rceil \tag{10}$$

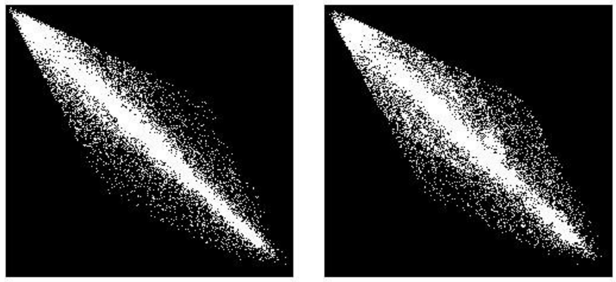
where $\lceil$ x $\rceil$ is called as the ceil function. Since $|x^{'} - y^{'}| = 3|x - y|$, the contrast increases after the mapping and hence the PSNR decreases.

Because of eqn 8, 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 grayscale values of transformed pixel pairs are restricted within a subdomain S_r defined as

$$S_r = \{(x', y') \mid x' \in [0, L), \; y' \in [0, L)\} \tag{11}$$

Where L is the number of graylevels. It can be seen from above constraint that the transform forms a domain of rhombus shape and can give a maximum embedding capacity of 0.5 bpp. This assumes that all pairs (x', y') fall within the rhombus, giving us to embed one bit for a pair of pixels (x, y). As this is the constraint for RCM based technique mentioned by Coltuc et al. [4], similar constriants are there for all techniques based on difference expansion as mentioned by Tian [8], Alattar [1] and Sachnev et al. [12]. Although the embedding capacity offered in some recent publications



1: Illustration of computation of embedding capacity from pair-wise co-occurance matrices for *Lena* image for displacements (0,1) and (1,2).

are more than 0.5 bpp and they go close to 1bpp in single iteration [7, 12, 6], still they pose a similar constraint for the data set as elligible for data embedding. This effectively means that they would form a geometric subdomain which would restrict the capacity.

3. PROPOSED SCHEME

We work on the similar mathematical relationship as the one mentioned in the Coltuc's paper [4], but in the above technique, one has to go to the actual data (image) and work out on each and every portion of image with different constraints as mentioned above, and finally we get the embedding capacity of the image. We here propose a scheme wherein the embedding capacity of the data set (image) can be calculated by the statistical parameters computed from the image.

1. *Co-occurance Matrix:* The concept is to calculate the pair-wise co-occurance matrix (this is slightly different from the conventional defination of co-occurance matrix where a particular pixel is considered in two different pairs for a given displacement while computing the entries in the matrix. In the current description, a pixel is considered only once for a given displacement i.e, none of the pixel pairs are overlapping in our defination) of the given data for different displacement between the pixel pair (as per given transform eqaution). Normally the displacement between pixel to pixel (pixel pair) is considered as $(1, 0)$, but for few highly textured images, better embedding capacity is observed for different distances along different directions.
   Implementation is done for all pairs of pixels and for a given displacement we could obtain a co-occurance matrix of rhombus type (domain $S \in [0, L]$ x $[0, L]$)), that satisfies the constraint eqn 11. The same concept can be extended to generalized transform as mentioned in Alattar [1]. The co-occurance matrix for such type of m-tuple data would be a co-occurance tensor for $(m-1)$ displacements with respect to the first pixel. Although all subsequent analysis carry forward, we explain for only m=2.

2. *Embedding and Decoding Algorithm:* As far as the concept is concerned, it can be applied to any arbitary dataset and one can use any technique for data embedding as mentioned previously. The implementation

1: Embedding capacity calculated from co-occurance matrix for *Lena* image for different displacements.

Displacement	Embedding Capacity(bits)	bpp
(0,1)	29,568	0.4512
(0,2)	26,100	0.3982
(0,3)	23,272	0.3551
(1,1)	28,452	0.4341
(1,2)	24,782	0.3781
(1,3)	21,896	0.3341
(2,1)	31,372	0.4787
(2,2)	29,234	0.4460
(2,3)	27,018	0.4122
(3,1)	29,288	0.4469
(3,2)	26,046	0.3974
(3,3)	23,478	0.3582

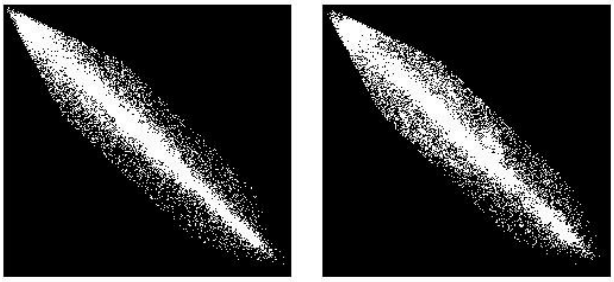
for RCM based watermarking proposed by Coltuc [4] is used in our study. The embedding and decoding algorithms follow equations 9 and 10, with a constraint of equation 11.

3. *Embedding Capacity:* Considering equations 9,10 and 11, we plot the co-occurance matrix of an image for different displacements and then adding up all the points in the rhombic region gives us the overall embedding capacity (the final embedding capacity will be obtained by substracting the payload from the total embedding capacity). Thus the embedding capacity is the cardinality of the set of points in the co-occurance matrix within the constraint region. This is illustrated in Fig 1. The payload includes the LSB's of pair of pixels which violate constraint equation 11.
Let A be the total number of pairs, and let E be the total pairs with embedded information (i.e E is the total embedding capacity for the image). Besides the payload, the LSB of the first pixel of the other $A-E$ pairs is to be stored, that is only $2E-A$ bits is the final space for watermark to be embedded. The final embedding capacity for the image is calculated as:

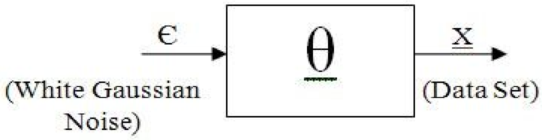
$$C = \frac{2E-A}{2A} bpp. \tag{12}$$

To ensure that there is space for embedding, it is to be noted that a practical embedding capacity will exist only till the time, the total embedding capacity $E > A/2$.

For the co-occurance matrices of different direction and distances, we consider the one which provides the maximum embeding capacity. The particular displacement which corresponds to the maximum embedding capacity is chosen to select the pixel pair (x, y). Considering above combinations, we proceed towards actual watermarking as per steps mentioned in embedding and decoding part. Table:1 shows the embedding capacity for *Lena* image for various choices of displacement for pixel pairing. For this particular image, the highest embedding capacity is obtained for displacement (2,1) which is chosen for watermarking purposes. For a different image, the corresponding pairing will be different.



2: Embedding capacity for displacements (0,1) and (1,2) as computable from the co-occurance matrices for *Lena* image for a control distortion factor $\delta = 50$.



3: An autoregressive model for the image.

4. *Incorporation of control distortion constraint:* For low data embedding applications, one can go for control distortion mechanism which improves the perceptual quality of image, as only those pixel pairs will be transformed which satisfy condition below [4]

$$|x - y| < \delta, \tag{13}$$

where δ is the threshold which reduces the embedding capacity of the image. This constriant allows only those pixel pairs which, in the co-occurance matrix, lie within the δ units of off-diagonal elements. This, along with constraint given in eqn 11 defines the embedding capacity as shown in Fig 2. From the cardinality of the set, the capacity can again be computed in the same way.

5. *Control distortion under image parameterization:* Till now we have been calculating the embedding capacity using a non-parametric method. This required the construction of co-occurance matrices. If a model for the image is given, it may be possible to compute the capacity directly from the model parameters θ.

For an autoregressive model illustrated in Fig 3, the output data vector will be $\underline{x}$. The embedding capacity could then be computed from the probabilty:

$$P\{[|x_i - x_{i+d}| < \delta \mid \theta] \bigcap [(0 \le x_i, x_{i+d} < L)]\} \tag{14}$$

where d is the chosen displacement.

6. *Multiple Embedding:* To increase the capacity of the image further, we suggest that one can go for multiple itreations of the watermarking at the cost of PSNR. This need not be done for the same displacement again. The best thing is to find the embedding capacity of the watermarked image using the same concept of co-occurance matrix and then based on the highest embedding capacity in the first pass for a particular displacement, one can watermark it again subsequently

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

using this embedded image as the input. This continues.

Theorem 1. *For a multipass embedding scheme, the cardinality of the set* $S = \{(x, y) \mid [\{|x - y| < \delta\} \bigcap \{0 \leq x, y < L\}]\}$ *always decreases after every pass of embedding.*

Proof. Since pixel pairs (x, y) which do not satisfy $|x - y| < \delta$ are not used for embedding, these pixel values do not change and hence none of the pairs in the dark region in Fig 2 can come inside the shaded region. On the contrary, the pairs within the shaded region due to change in LSB and the transformation(eqn 9) while embedding may come out of the range $|x - y| < \delta$. Hence the proof. $\square$

The iterative scheme of embedding stops when there is practically no further space for embedding watermark bits. Thus, although the over all embedding capacity increases, the capacity per individual iteration reduces. Thus multiple embedding could be done till the time no space is left for embedding. i.e. $E > A/2$ condition is no more satisfied.
Normally the embedding gets over by 5-6 iterations but, for highly textured images, the embedding capacity may get over earlier. Also the embedding capacity gets exhausted much earlier as we decrease the value of δ. This suggests that for high PSNR applications, the embedding capacity is very limited.

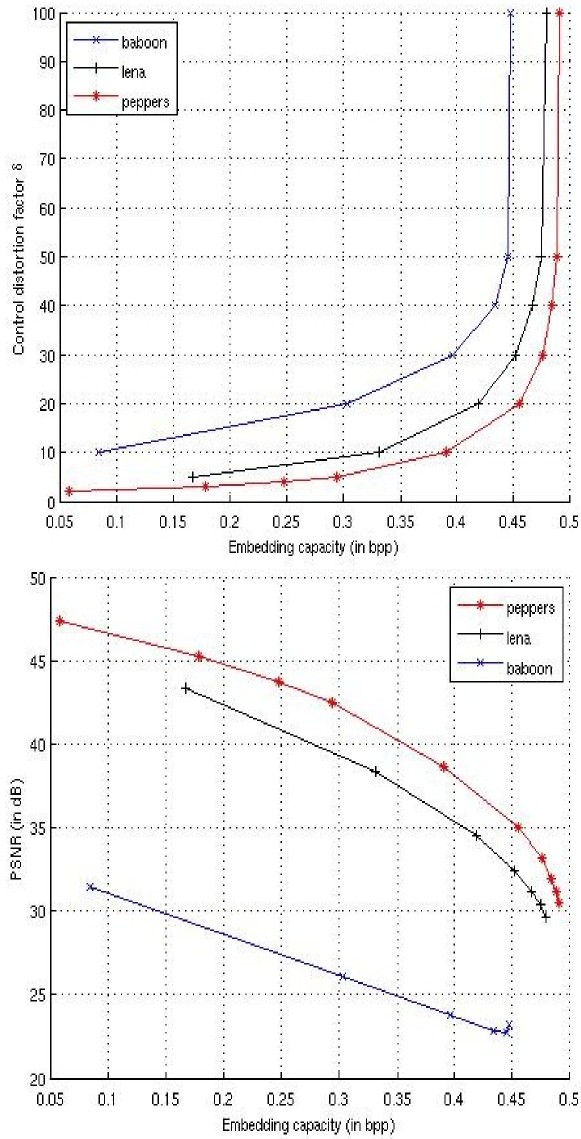
4. EXPERIMENTAL RESULTS

The presented concept was experimented on several commonly known images. We present results only for 3 standard images, as shown in Fig 4, for brevity. We study the following characteristics while embedding the watermark: how does the cover image appear after embedding, how does the PSNR change as the distortion δ and the number of passes increase, and how does the PSNR change as the amount of embedding is increased. In Fig 5 we display the results of embedding a watermark in *Lena* image under no control distortion condition for four different passes. Any distortion is barely visible in Fig 5(a) after the first level of embedding when the embedding load is 0.4787 bpp. After the second pass, the load goes upto 0.8957 bpp, but the distortion is still tolerable. After the third pass, the embedding load becomes 1.2392 bpp but the distortion becomes quite noticeable. After the fourth pass when the load becomes 1.4602 bpp, there is a significant distortion. Hence it may not be advisable to attempt to embed any more bits into the image (although some embedding capacity is still available in further iterations). It may be noted that as the embedded load increases, the cover image becomes more susceptable by attacks in terms of detectability of existance of watermark. We do not investigate this issue in this paper.

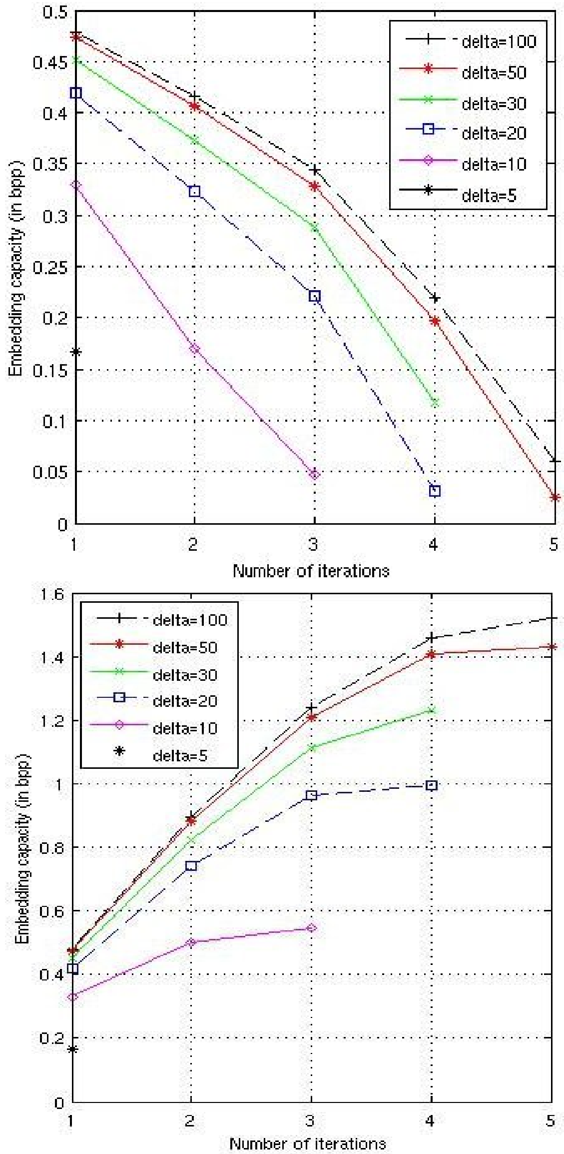
5: Watermarked copies of *Lena* image after multiple embedding a)single, b)twice, c)thrice, d) four times, without control distortion.

In Fig 6(a) we present the effect of varying the control distortion parameter δ during embedding. As δ is small, the cardinality of the set of points satisfying the embeddability criterion is quite small and hence the corresponding bpp is small. As δ increases, the capacity increases initially slowly and then rapidly to saturate toward a value close to 0.5 as suggested in eqn 12. This plot also shows that the capacity also depends on the textural contents of the cover image. For a predominantly low pass image *'Pepper'*, the capacity is higher than that of a slighty more busy picture *'Lena'*. For the very busy picture *'Baboon'*, the capacity is even lower. In Fig 6(b) we show the corresponding effect on the PSNR of the embedded cover image. Quite naturally, as the embedded load increases, the PSNR decreases and the PSNR gets more affected if the original cover image has very high frequency contents.

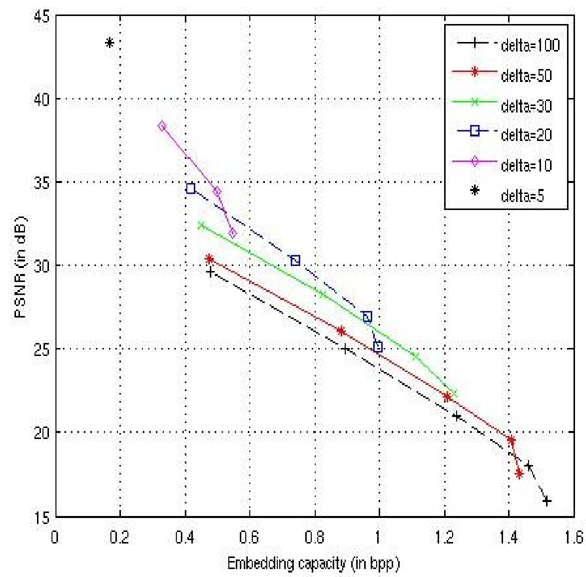
For Multiple iterations, the embedding capacity goes on reducing for further iterations as some of the pixel pairs which were in the area before watermarking (i.e satisfying the constriant) has now come out of the area (i.e. they do not satisfy the constraints now). This can be seen from Fig 7(a). This plot experimently verifies the statement in theorem 1. But the over-all (final) embedding capacity of the image increases as we add the embedding capacities of individual iterations. This can been in Fig 7(b). It can be clearly observed for this figure that as δ increases, once can run multiple passes to embed more watermak bits. For smaller choice of δ (and thus more stringent condition on output PSNR), one may not be able to continue that many iterations as the cardinalty of the admissibilty set approaches to



6: Results of single pass embedding. a) capacity for different values of δ and b) corresponding PSNR-bpp relationship.



7: Results of multipass embedding for *Lena* image for various values of control distortion parameter δ. a)Incremental gain in capacity per iteration, and (b) cumulative gain as iterations proceed.



8: Results of multi-pass embedding displaying bpp-PSNR relation for *Lena* image for various values of δ

zero very quickly. For example, when $\delta = 5$ only, all options to embed watermark get exhausted in the first pass itself. But how does this affect the PSNR value ?. This is plotted in Fig 8 as a function of embedded load (in bpp) for various values of δ and the corresponding multiple passes. This is an impotant plot as it summarizes all earlier plots for the cover image *Lena*.

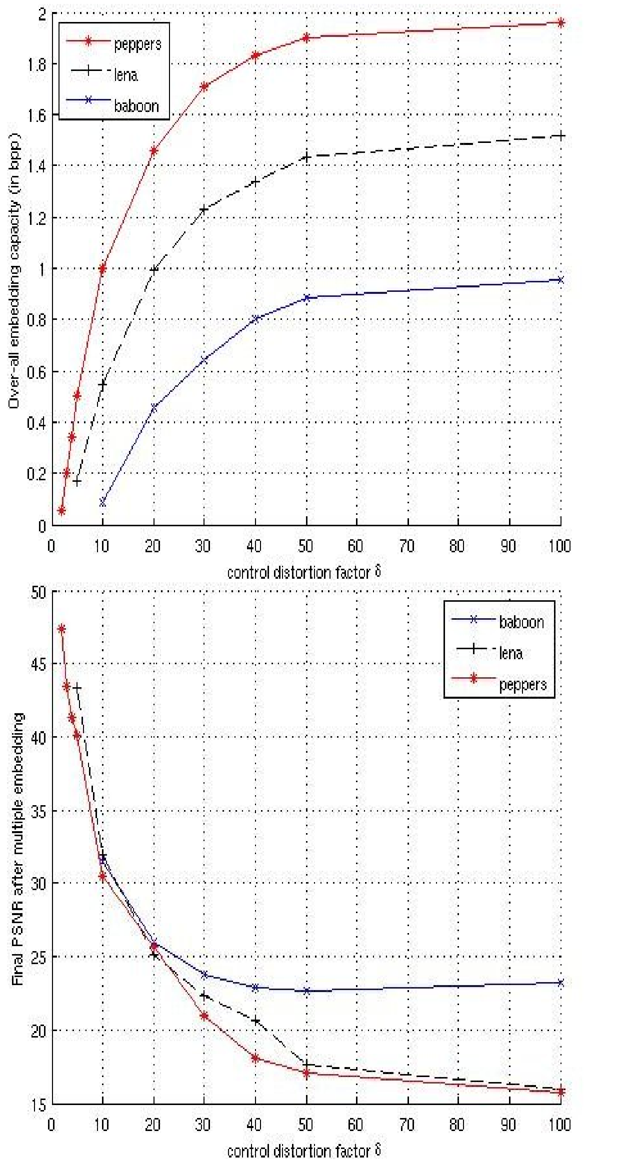
One can easily make out that the maximum embedding capacity for *lena* image in a single iteration is 0.478 bpp with a PSNR of around 29dB. The same embedding capacity can be obtained with a higher PSNR using multiple embedding and a better choice of δ. This can be seen from figure 8, where one can get a bpp of around 0.5 bpp with a reasonably good PSNR of 34 dB. The value of δ chosen for this is 10. One can thus decide a wise bpp-PSNR combination with a better choice of control distortion factor δ to obtain it. This plot also shows that a single pass embedding scheme is not PSNR-efficient. One should rather use multiple passes to embed with an appropriate δ value. Needless to say that the embedding capacity is also dependent on the textural details of the cover image. In Fig 9 we show this dependance for the three different images *Peppers*, *Lena* and *Baboon*, each with increasing textural details. Fig 9(a) shows that as δ increases, the capacity increases in all cases, but the increase is more for the less busy images. Similarly Fig 9(b) shows that there is an equivalent penalty in PSNR as δ increases.

5. ACKNOWLEDGMENTS

We are thankful to Rishabh Iyer at IIT Bombay for helping in the implementation of the RCM algorithm [4].

6. CONCLUSIONS

A novel technique of calculating the embedding capacity from the given data without having to do actual watermarking has been suggested in this paper. The proposed concept is equally applicable and with slight modification extendable to other difference expansion based schemes also [5, 8,



9: Comparision of results for different cover images. a) capacity utilization, and b) PSNR values for different values of control distortion factor δ.

12]. The results so obtained exactly match the results of actual embedding capacity after watermaking. An experimental analysis shows that the multipass embedding not only improves the embedding capacity (bpp value) but also improves the PSNR for appropriate selection of control distortion parameter δ.

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] I. Cox. *Digital Watermarking and Steganography.* Morgan Kaufmann,Second Edition, New York, 2008.

[3] D.Coltuc and J.-M. Chassery. High capacity reversible watermarking. *Proceedings of the IEEE International Conference on Image Procesing ICIP*, pages 2565–2568, 2006.

[4] D.Coltuc and J.-M. Chassery. Very fast watermarking by reversible contrast mapping. *IEEE Signal processing Letters*, 14(4):255–258, 2007.

[5] D.M.Thodi and J.J.Rodriquez. Expansion embedding techniques for reversible watermarking. *IEEE Trans. Image Processing*, 16(3):721–730, 2007.

[6] L. et. Reversible image watermarking using interpolation technique. *IEEE Trans Information Forensics and Security*, 5(1)(187-193), 2010.

[7] W.-L. T. et. Reversible data hiding based on histogram modification of pixel differences. *IEEE Trans. Circuits Systems for Video Technology*, 19(906-910), June 2009.

[8] J.Tian. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Systems for Video Technology*, 13(8):890–896, August 2003.

[9] C.-T. Li. Reversible watermarking scheme with image-independent embedding capacity. *IEE Trans on Vision and Image Processing*, 152(6)(779-786), 2005.

[10] 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.

[11] M.U.Celik and G.Sharma. Lossless generalized-lsb data embedding. *IEEE Transactions on Image Processing*, 14(2):253–266, 2005.

[12] V. Sachnev. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Systems for Video Technology*, 19(7):989–999, July 2009.