Watermarking Capacity Control for Dynamic Payload Embedding

Hussain Nyeem[1], Wageeh Boles[2], and Colin Boyd[2,3]

[1]Khulna University of Engineering and Technology (KUET), Khulna, Bangladesh
[2]Queensland University of Technology (QUT), Brisbane, Australia
[3]Norwegian University of Science and Technology (NTNU), Trondheim, Norway
h.nyeem@kuet.ac.bd, w.boles@qut.edu.au, colin.boyd@item.ntnu.no

Abstract. Despite significant improvements in capacity-distortion performance, a computationally efficient capacity control is still lacking in the recent watermarking schemes. In this paper, we propose an efficient capacity control framework to substantiate the notion of watermarking capacity control to be the process of maintaining "acceptable" distortion and running time, while attaining the required capacity. The necessary analysis and experimental results on the capacity control are reported to address practical aspects of the watermarking capacity problem, in dynamic (size) payload embedding.

Keywords: Capacity control, dynamic payload, embedding capacity, fragile watermarking

1 Introduction

Digital image watermarking has drawn much attention for improving the embedding capacity-distortion performance [1–5, 9–18, 25, 27–29]. Particularly, recent fragile watermarking schemes aim at achieving high capacity with the lowest possible distortion. (Fragile watermarks, by definition, become invalid for any possible modification in a watermarked image.) In some fragile watermarking applications (*e.g.*, annotation) the payload size (*i.e.*, watermark plus any side information) significantly varies [21]. We call such a payload *dynamic*, which requires varying capacity with a relatively high upper bound. Achieving this requirement, however, is challenging (specially under the perceptual constraints of images) and requires capacity control.

Watermarking *capacity control* is the process to achieve the required capacity while maintaining a low level of distortion. Ideally the process should minimise the distortion. However, when the payload is dynamic this may lead to an unacceptable running time and so in this paper we study the notion of capacity control to be the process of maintaining "acceptable" *distortion* and *running (or computational) time*. (The term "acceptable" means to be minimum, but its level may vary with the applications.) Here, distortion is the degree of perceptual degradation incurred by the embedding function, and running time [6] is measured by the number of machine-independent operations executed.

In watermarking research, the best capacity-distortion performance are shown by the reversible schemes [1–5, 9–15, 18, 25, 27–29]. These fragile schemes introduce an invertible distortion in a watermarked image. Tian [28] pioneered the DE scheme addressing the low capacity and/or noticeable visual artefact problems of earlier feature compression based reversible schemes [2, 8, 9]. Tian's scheme is later generalised by Alattar [1], and then improved using the sorting of pixel-pairs by Kamstra and Heijmans [12] for higher embedding capacity. Kim *et al*. [13] improved Tian's scheme using the simplified location map to reduce the overhead data.

Additionally, Ni *et al*. [18] introduced the histogram shifting (HS) scheme, which does not require any location map (but a pair of peak/zero points in the histogram). Ni *et al*.'s scheme is later improved by using the difference-histogram [14], rhombus predictor and sorting [25], adaptive and multilevel embedding [15, 24] for better perceptual quality and higher embedding capacity. Reversible contrast matching (RCM), another invertible transform, -based scheme is proposed by Coltuc and Chassery [5] and extended by Chen *et al*. [3]. Thodi and Rodríguez [27] combined the HS and DE techniques and introduced the prediction error expansion (PEE) scheme, which was later improved by Hu *et al*. [11] for better capacity-distortion performance.

However, the above (and many others) DE-, HS-, RCM- and PEE-based reversible schemes usually have an inefficient capacity control and thus may not be suitable for dynamic payload embedding. They often require a *recursive* (or *multi-level*) embedding in support of capacity control. A recursive embedding re-embeds any remaining part of a payload recursively in a watermarked image until the required capacity is achieved. The repeated alterations of pixels may not always incur more distortion if the same bit-plane(s) is used for re-embedding, but they are more likely to significantly grow the running time of the schemes. To demonstrate the need for an efficient capacity control, and to thus address a novel aspect of the embedding capacity problem, form the motivation of the research reported in this paper.

As the main contribution, this paper presents an efficient capacity-control framework. It is more than challenging to develop a unified capacity control framework for watermarking due to the variety of its techniques and applications. A case of fragile watermarking schemes is therefore considered for dynamic payload embedding, where we often face the dilemma of limiting the size of payloads or sacrificing the performance of an embedding scheme. In this paper, we determine the impact of the inefficient (*e.g*., recursive) capacity control of watermarking schemes on their overall performance. Thereby, we validate the proposed framework with the asymptotic analysis and necessary experiments of watermarking schemes having different capacity control.

The rest of the paper is organised as follows. A new capacity control framework is proposed in Sec. 2. In Sec. 3, asymptotic analysis of capacity control of watermarking schemes are given in light of the proposed framework. Experimental results are discussed in Sec. 4 followed by the conclusions in Sec. 5.

2 A New Capacity Control Framework

In this section, we present a capacity control framework for efficient embedding of dynamic payload. (We adopt necessary notations from [20].) As discussed in Sec. 1 and shown in Fig. 1 (*existing scenario*), current capacity control ideally aims at minimizing the distortion only and thus lacks consideration of the running time. Here, the Hu *et al*. scheme [11] (that we will analyse in Sec. 3.2 to validate our framework) is a prominent example that fits the existing scenario. Therefore, our proposed framework (incorporating the *extended scenario* in Fig. 1) aims to ensure the attainment of the required capacity with both the least possible distortion and running time simultaneously. For an input image and payload, the framework seeks a suitable capacity parameter setting with possible user intervention to update the predefined thresholds (or to reconsider the inputs/embedding scheme, in a worst case scenario). The general steps are discussed below.

Let an embedding function, $E(\cdot)$, embeds a watermark, W and side information, S_{info}, in an input image, I such that $\bar{I} \leftarrow E(I, payload)$. Here, $\bar{I}$ is the watermarked image and the *payload* is computed by a concatenation function, $Concat(\cdot)$, *i.e.*, $payload \leftarrow Concat\left(W, S_{info}\right)$. To find the least possible distortion, *Dist* and running time, E_t, the set of capacity parameters *par* and the thresholds (T_1, T_2) for (*Dist*, E_t) are initialized with their minimum possible values. With that setting, a capacity estimation function, $Est(\cdot)$ computes the total (available) capacity, C_t. The required capacity, C_p is determined using $Size(\cdot)$ that returns the bit-length of its input such that $C_p \leftarrow Size(payload)$. Until the C_p is achieved, *par*, T_1 and T_2 are updated as shown in Fig.1.

The influence of increasing C_p on *Dist* and E_t. can be controlled by the capacity parameter, *par*, and the thresholds T_1 and T_2. An efficient updating of *par* is here crucial for the capacity control to minimize E_t. For example, based on the difference between C_t and C_p, an adaptive update of *par* may significantly minimize the time needed to reach the required capacity level. Bearing this in mind, we define the capacity control efficacy in Def.1 below. Here, we consider only C_p, since C_t accounts for C_p (possibly with an increasing *Dist* and E_t for $C_t \geq C_p$).

Definition 1. (capacity control efficacy). *An embedding function, $E(\cdot)$ is said to have an efficient* capacity control*, if it ensures the attainment of the required capacity, C_p with minimum possible distortion, Dist and computation time, E_t, where Dist and E_t grow with the minimum possible amount/step as C_p grows.*

In order to demonstrate the viability of Def. 1, and thus to determine the capacity control efficacy of $E(\cdot)$ for dynamic payload embedding, we pose the following questions: (*i*) what are the *Dist* and E_t values of $E(\cdot)$ for the lower bound of the dynamic payload? and (*ii*) at what rate should *Dist* and E_t of $E(\cdot)$ be changed for the increasing C_p? Analysing watermarking schemes in light of these questions would lead to the conclusion that the lower the (quantitative) values of the said parameters, the higher the efficacy of the schemes.

3 Capacity Control Analysis

We analyse and experiment with the Nyeem, Boles, and Boyd (or NBB) scheme [22,23] and Hu, Lee, and Li (or HLL) scheme [11] below as to determine the performance of their capacity control for dynamic payload embedding. The choice of the HLL scheme is made as it is a prominent watermarking scheme having capacity control that closely represents the existing capacity control scenario, as mentioned in Sec. 2. We presented the NBB scheme in [22, 23] that follows the capacity control framework proposed in

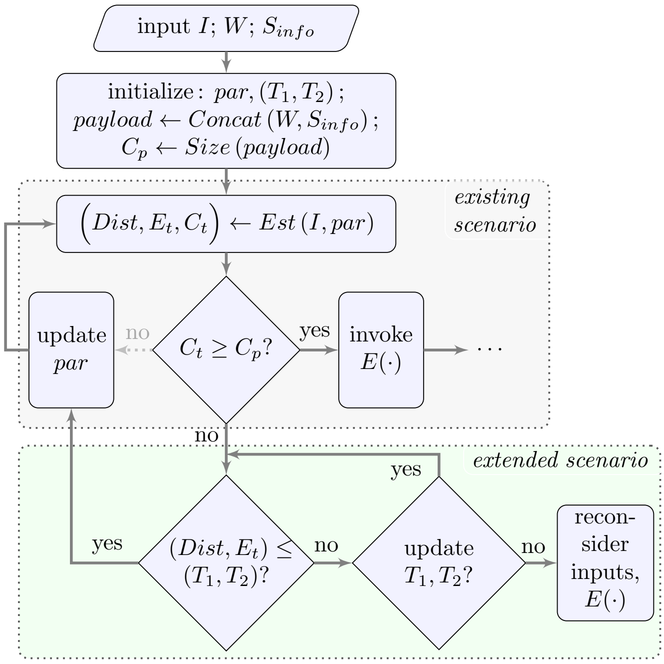

Fig. 1. Flow-chart of the proposed capacity control framework for dynamic payload embedding [19].

this paper. Since, both schemes have capacity control, they can be used for dynamic payload embedding.

3.1 The NBB Scheme Analysis

NBB scheme [22, 23] is proposed to provide continuous security protection and to minimize the legal-ethical issues of medical images. That scheme embeds payloads in the LSB (least significant bit) planes of the border pixels of input images. A greater capacity control is targeted in terms of N_{BW} and N_{LSB} with their thresholds T_{BW} and T_{LSB} respectively, as shown in Fig. 2. Here, N_{BW} is the number of pixels in a given border width, N_{LSB} is the number of LSB-planes, and c_t is calculated using $C_{total} = 2N_{BW} \times (r + c - 2N_{BW}) \times N_{LSB}$. We note that Fig. 2 does not explicitly show any consideration for E_t as shown in the proposed framework in Fig. 1. Because the NBB scheme does not consider recursive embedding, its running time always remains in $O(n)$. The capacity control running time of NBB embedding function is $n \times (c_5 + c_6 + c_7 + c_8)$, where c_5 to c_8 are time constants for the steps shown in Fig. 2.

3.2 The HLL Scheme Analysis

The HLL scheme [11], on the other hands, expands the (median) prediction errors (*i.e.*, $p_e = x - \hat{x}$) using classical DE rule (*i.e.*, $p'_e = 2p_e + b$) and its variant (*i.e.*, $p'_e = 2p_e - b$). (Where, x and $\hat{x}$ are original and predicted versions of the pixels, p_e and p'_e are original and expanded versions of the errors, respectively and b is the watermark bit.) Thereby,

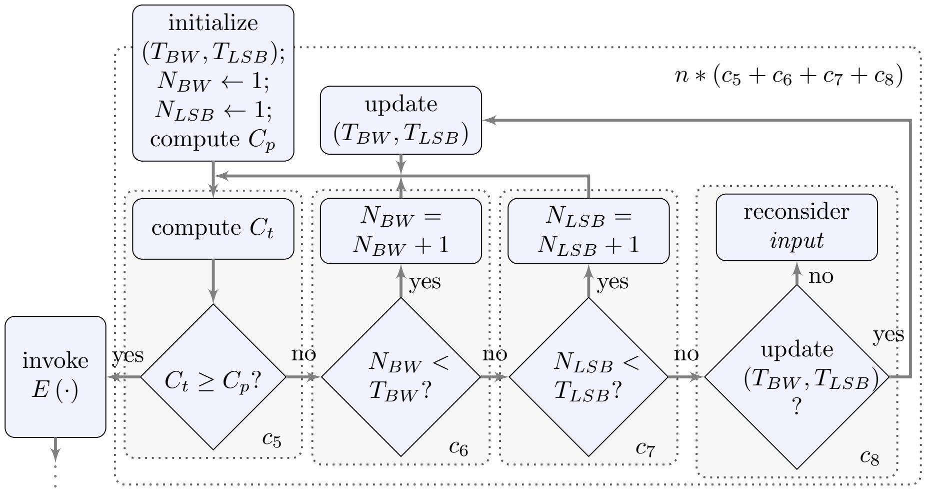

Fig. 2. Capacity control flow-chart of NBB scheme [19].

an interleaving approach (*e.g.*, adding a bin first rightward, then leftward, and so on or *vice-versa* for an embeddable region) is introduced for capacity control, considering a first round embedding. This consideration, however, is also suggestive of possible successive embedding rounds. To demonstrate the consequences of such embedding, we perform an asymptotic analysis of the HLL scheme. to determine its rate of growth of running time.

HLL capacity control has two main parts: *OUF* (*over-/under-flow*) *map construction* and *scanning* as shown Fig. 3. For the given I and C_p, the OUF location map, M is constructed recursively for each *pixel*—a vector for the current pixel and the $flag$—a flag-bit for HS direction (*e.g.*, 0 for the left). The JBIG (Joint Bi-level Image Experts Group) compression is then used: $\hat{M} \leftarrow JBIG(M)$, where $\hat{M}$ is JBIG compressed version of M. (For more details, see [11].)

The OUF map construction and scanning have the running time of $c_4 n \times (c_1 n + c_2 \log n)$ and $c_3 n$, respectively, where n is input size—the total number of pixels, and c_1 to c_4 are time constants—the fixed time period taken by the set of operations. So, the capacity control running time becomes $c_1 c_4 n^2 + c_2 c_4 n \log n + c_3 n$ leading to an asymptotic upper bound $O(n^2)$ (considering the JBIG running time is $O(\log n)$ as being arithmetic coding based [26]). With the running time of $E(\cdot)$ in $O(n)$, the HLL scheme's embedding performance depends on its capacity control, leading to an overall upper bound of $O(n^2)$. So, for a k-round embedding, the running time will be in $O(n^{2k})$, where even with a small value, k will severely impact on the HLL scheme's overall performance.

4 Experimental Results and Discussion

We examined the performance of the schemes in question [11,22] with varying payload size. We performed several experiments using 150 test images (from [7]). An example of a few test-set images are shown in Fig. 4. As shown in Fig. 5 (1^{st} row), unlike NBB scheme, where the capacity control running time remains steady and much lower,

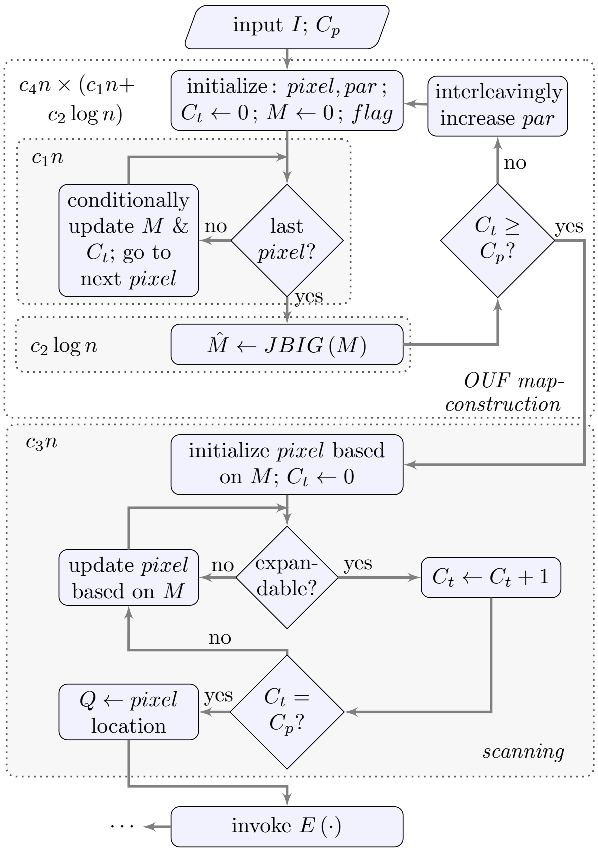

Fig. 3. Capacity control flow-chart of the HLL scheme. (Dotted-blocks indicate their approximate worst-case running time [19].)

(a) (b) (c) (d)

Fig. 4. Sample test images of size $512 \times 512 \times 8$: (a) Aerial, (b) Mandrill, (c) Stream and bridge, and (d) Man. (Available here [7])

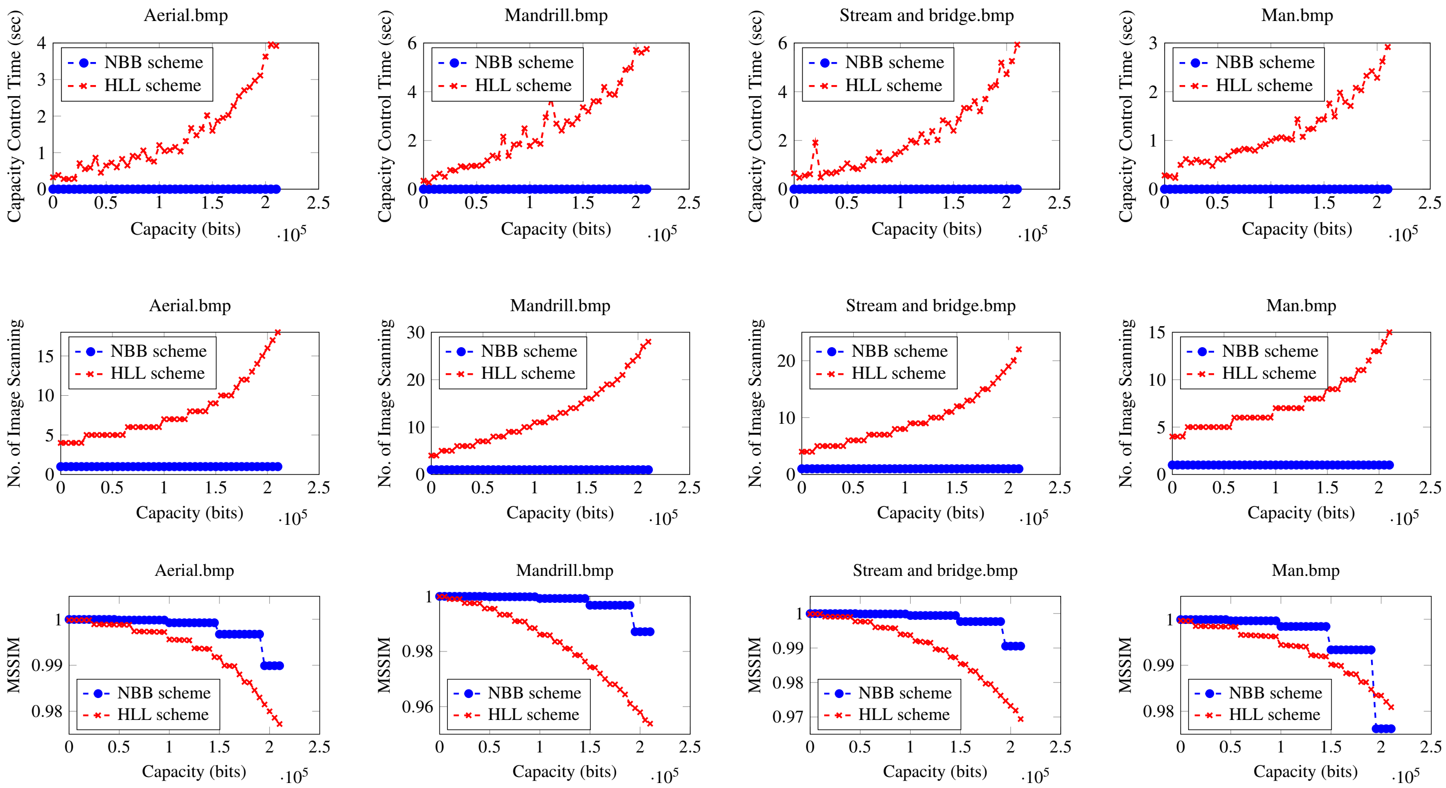

Fig. 5. Evaluation and comparison of capacity control efficacy for *Aerial*, *Mandrill*, *Stream and bridge*, and *Man*. From top, capacity control time (1^{st} row), number of scanning the input image (2^{nd} row), and MSSIM (3^{rd} row).

implementation dependent, and we used MATLAB (7.14.0.739) and an Intel Core i5 3.2GHz CPU for our experiments.

Additionally, a step pattern in the performance variation is evident for the HLL scheme (Fig. 5: 2^{nd} & 3^{rd} row) and for the NBB scheme (Fig. 5: 3^{rd} row). This means that their performance, which is dependent on the payload size, remain unchanged until the capacity condition (*i.e.*, $C_t \geq C_p$) is satisfied. Otherwise, respective capacity parameters are increased with a step (up/down) pattern in their performance curves. Unlike the HLL scheme, the equal step-sizes also mean that each increment of capacity parameters consistently gives a fixed amount of capacity increment for the NBB scheme. We note that we kept N_{BW} fixed to 25 and varied N_{LSB} from 1 to 5, because we have shown in [22] that increasing the number of LSBs is more effective for meeting higher capacity requirements. We also considered the payload size of 1 Kbits to 215 Kbits.

Moreover, Fig. 5 (3^{rd} row) shows Mean Structural SIMilarity (MSSIM) of the watermarked images for the schemes in question. MSSIM, a particularly designed image quality metric, measures the local similarity of perceptual contents thus it mainly accounts for the changes in perceptually significant information.The general formulation of MSSIM [30] is given below in (1).

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \tag{1}$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

where, x_j and y_j are the image content at j-th local window and their structural similarity index, $SSIM(x_j, y_j)$ is computed using (2). Here, μ_x and μ_y are the average values of x and y, and σ_x^2 and σ_y^2 are the variance of x and y, respectively; σ_{xy} is the covariance of x and y; and $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ are two variables to stabilize the division with weak denominator for the L dynamic range of the pixel values. The default values of the weight factors, k_1 and k_2 are set to 0.01 and 0.03, respectively.

This MSSIM curves in Fig. 5 (3^{rd} row) suggest that both schemes would have more distortion for increasing the payload size further. This would also drastically grow the running time of HLL scheme demonstrating the possible (severe) impact of multilevel embedding. This is unlike the NBB scheme, where its running time would remain nearly steady.

5 Conclusions

We have discussed some practical aspects of watermarking capacity and pointed out that embedding of increasing size payload would contribute to the exponentially increasing computational overheads. Thus, a watermarking scheme could eventually be less efficient for an application. Addressing this problem, we have presented a heuristically designed framework for efficient capacity control. We examined the efficiency of our proposed framework by an asymptotic analysis of the HLL scheme and NBB scheme, where these schemes closely represent the existing and proposed frameworks of capacity control, respectively. We also verified the efficiency of the proposed framework by analysing the performance variations (resulting from the dynamic payload embedding) of HLL and NBB schemes.

We argue that the existing capacity control is more interpreted as a trade-off between capacity and distortion requirements. However, failure to also consider the running time may render a scheme less practicable for an application, especially if dynamic payload is a requirement. Therefore, addressing the questions posed in Sec. 2, we have shown that the capacity control of NBB scheme outperforms that of the HLL scheme. The asymptotic analysis and experimental results demonstrate further the consequences of an inefficient capacity control and thus validate the efficiency of the proposed capacity control framework. We note that the given results are implementation dependent, and possible optimized implementation could give better results than what we found. However, the trends shown in the graphs (in Fig. 5) will apply for any implementation.

Moreover, since the HS-, DE-, RCM- and PEE-based schemes usually have a similar capacity control principle like the HLL scheme, they should have more or less similar effect on the running time, for increasing payload size as well as for recursive embedding. Although different watermarking principles and application requirements make it a more challenging task, the proposed capacity control framework may be reduced to a generalized form in future.

6 Acknowledgement

The authors would like to Thank to Lance De Vine (QUT) for JBIG help.

This paper presents a part of the work of the first author's PhD project and thus contains some of the material available in that thesis [19].

References

1. Alattar, A.M.: Reversible watermark using difference expansion of quads. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 377–380. IEEE (2004)
2. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-lsb data embedding. IEEE Transactions on Image Processing 14, 253–266 (2005)
3. Chen, X., Li, X., Yang, B., Tang, Y.: Reversible image watermarking based on a generalized integer transform. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 2382–2385. IEEE (2010)
4. Coatrieux, G., P., W., Cuppens, B.N., Cuppens, F., Roux, C.: Reversible watermarking based on invariant image classification and dynamic histogram shifting. IEEE Transactions on Information Forensics and Security 8, 111–120 (2013)
5. Coltuc, D., Chassery, J.M.: Very fast watermarking by reversible contrast mapping. IEEE Signal Processing Letters 14(4), 255–258 (2007)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms (2^{nd} Ch.). 3^{rd} Ed., MIT Press (2011)
7. The USC-SIPI image database, `http://sipi.usc.edu/database`
8. De Vleeschouwer, C., Delaigle, J.F., Macq, B.: Circular interpretation of bijective transformations in lossless watermarking for media asset management. IEEE Transactions on Multimedia 5, 97–105 (2003)
9. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding-new paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing 2002(1), 185–196 (2002)

10. Guo, X., Zhuang, T.G.: A region-based lossless watermarking scheme for enhancing security of medical data. Journal of Digital Imaging 22, 53–64 (2009)
11. Hu, Y., Lee, H.K., Li, J.: DE-based reversible data hiding with improved overflow location map. IEEE Transactions on Circuits and Systems for Video Technology 19, 250–260 (2009)
12. Kamstra, L., Heijmans, H.J.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Transactions on Image Processing 14, 2082–2090 (2005)
13. Kim, H.J., Sachnev, V., Shi, Y.Q., Nam, J., Choo, H.G.: A novel difference expansion transform for reversible data embedding. IEEE Transactions on Information Forensics and Security 3, 456–465 (2008)
14. Kim, K.S., Lee, M.J., Lee, H.Y., Lee, H.K.: Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recognition 42, 3083–3096 (2009)
15. Lee, S., Yoo, C.D., Kalker, T.: Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Transactions on Information Forensics and Security 2, 321–330 (2007)
16. Lin, C.C., Tai, W.L., Chang, C.C.: Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognition 41, 3582–3591 (2008)
17. Ni, Z., Shi, Y.Q., Ansari, N., Su, W., Sun, Q., Lin, X.: Robust lossless image data hiding designed for semi-fragile image authentication. IEEE Transactions on Circuits and Systems for Video Technology 18, 497–509 (2008)
18. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology 16(3), 354–362 (2006)
19. Nyeem, H.: A digital watermarking framework with application to medical image security. Ph.D. thesis, QUT, School of Electrical Eng. and Computer Science, Australia (July 2014)
20. Nyeem, H., Boles, W., Boyd, C.: Developing a digital image watermarking model. In: Proceedings of 13^{th} International Conference on Digital Image Computing: Techniques and Applications, pp. 468–473. IEEE, Piscataway (2011)
21. Nyeem, H., Boles, W., Boyd, C.: A review of medical image watermarking requirements for teleradiology. Journal Digital Imaging 26, 326–343 (2013)
22. Nyeem, H., Boles, W., Boyd, C.: Utilizing least significant bit-planes of roni pixels for medical image watermarking. In: Proceedings of 15^{th} International Conference on Digital Image Computing: Techniques and Applications, pp. 1–8. IEEE, Piscataway (2013)
23. Nyeem, H., Boles, W., Boyd, C.: Content-independent embedding scheme for multi-modal medical image watermarking. BioMedical Engineering Online 14(7), (2015)
24. Peng, F., Li, X., Yang, B.: Adaptive reversible data hiding scheme based on integer transform. Signal Processing 92, 54–62 (2012)
25. Sachnev, V., Kim, H.J., Nam, J., Suresh, S., Shi, Y.Q.: Reversible watermarking algorithm using sorting and prediction. IEEE Transactions on Circuits and Systems for Video Technology 19, 989–999 (2009)
26. Simpson, M., Biswas, S., Barua, R.: Analysis of compression algorithms for program data. University of Maryland (2003)
27. Thodi, D.M., Rodríguez, J.: Expansion embedding techniques for reversible watermarking. IEEE Transactions on Image Processing 16, 721–730 (2007)
28. Tian, J.: Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology 13(8), 890–896 (2003)
29. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Processing 89, 1129–1143 (2009)
30. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)