

REVERSIBLE DATA HIDING SCHEME VIA RECURSIVE CODES FOR GRAYSCALE AND BINARY COVERS

B.S.H. Shayeez Ahamed¹, Kumar.G.J²

^{1,2}Assistant Professor, Department of Computer Science & Engineering, MITS Madanapalle, Andhra Pradesh, India

Abstract

Data hiding is widely used to hide information into covers such as audio, video and image files. If the embedded information is extracted without causing visible distortion to original cover, it is known as Reversible Data Hiding (RDH). In reversible data hiding (RDH), the original cover can be losslessly restored after the embedded information is extracted. Many approaches came into existence including rate-distortion bound approach. Recursive construction is one of the approaches for rate-distortion bound. This paper presents RDH approaches via recursive codes for binary covers and steganography for grayscale covers. We built a prototype application to demonstrate the efficiency of both models. The empirical results reveal that the proposed RDH schemes are effective and used for real time applications.

Keywords – Reversible Data Hiding, Recursive Code Construction, Watermarking and Steganography

1. INTRODUCTION

Data hiding has been around for long time which is a technique used to embed information into covers such as video, audio, and image files. It has many real time applications such as covert communication, media notation, integrity authentication and copyright protection etc. LSB (Least Significant Bit) is being used by many existing data embedding methods. The embedding process in LSB results in distortion. It does mean that it is not possible to reconstruct original cover image. Thus it is not possible to reconstruct the original cover from marked cover. Nevertheless, in many real time applications like law forensic, military imagery, medical imagery etc. the deterioration of original image is not acceptable. This is the case where RDH (Reversible Data Hiding) is mandatory. RDH is nothing but lossless hiding of data which enables the system to recover original cover without distortion. The RDH technique was first introduced by Fridrich and Goljan [1] where the process of embedding is divided into three states namely extracting features of image, compressing the features, and embedding messages. The general RDH framework at sender side is as given in figure 1.

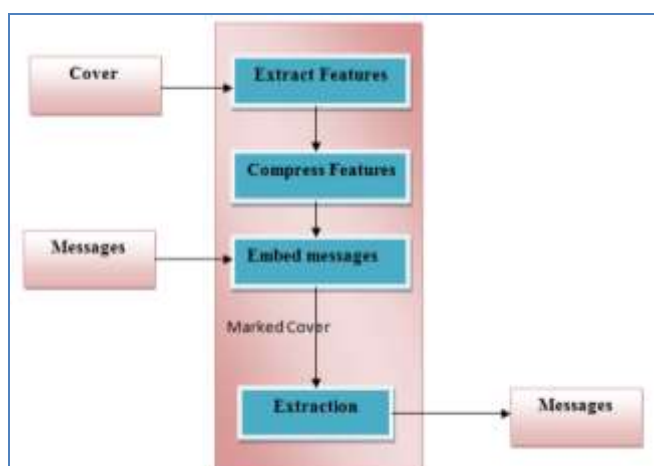


Fig. 1 –Framework of RDH

As can be seen in fig. 1, it is evident that lossless extraction of features of the input image which are compressible is done in first stage. In the second stage the extracted features are compressed which is also lossless in nature. This operation saves required space in the image for payloads or the messages to be embedded (secret hiding). In the third stage the messages are embedded into the sequence of features to produce a marked cover. In other words, the marked cover is generated when the feature of original cover is compressed and the messages are embedded into the space obtained losslessly. It does mean that the original features are replaced by the compressed features plus the images embedded in order to form marked cover. Thus it is possible to obtain the original cover once the messages are extracted by the receiver from cover image. The original cover image and the cover image after extraction must be same. This is the proof of RDH.

There were many researches on RDH which are described here. Fridrich and Goljan [1] suggested certain features are to be obtained in case of both spatial and JPEG images after exploiting image's global characteristics. For instance they suggested to use features like texture complexity and coefficients middle frequency DCT (Discrete Cosine Transform) respectively for spatial and JPEG images. Celik *et al.* [2] predict multiple LSB planes and enhances the scheme proposed in [1]. However, many other researchers used the idea presented in [1] for RDH into binary images [3], [4] or videos [5], [6].

Later DE (Difference Expansion) was introduced by Tian [7] which is aimed at extracting longer feature sequences and compressed. In DE the features are nothing but the differences between the neighbouring pixels. By multiplying the differences by 2 and then using LSBs of such differences are used to embed secret messages into image. Later on the DE approach has been adopted many researchers [8], [9], [10], and [11] to vector pixels, to reducing size of location

map, and to prediction errors respectively. HS (Histogram Shift) is another method used for RDH. As per this, the compressible features of image are represented as histogram. The compression technique to be used with HS was proposed by Ni *et al.* [12]. There are similar researches that make use of HS for RDS [13], [14]. There is a common feature in both DE and HS use special compression approaches for introducing distortion to original cover. In contrast to this [1] uses binary feature sequence and the compression operation does not introduce distortion. Based on these differences RDH can be divided into two types namely type I and type II. The former uses binary features and a generic compression algorithm. The techniques used in [1]-[6] come under this category. The latter non binary features and compression is performed using some specific approach such as DE [7]-[11] and HS [12]-[14]. According to Kalker and Willems the receiver of the image can reconstruct the cover from marked cover. This enables the sender to compress the cover as per certain condition which is the reason for the feasibility and efficiency of recursive construction.

In [15], the authors improved recursive construction with both conditional compression and conditional embedding in rate-distortion bound fashion. However, this has three limitations. The first one is that it is unable to reach rate-distortion bound. The second one is that the codes can't approach to a maximum embedding rate. The third one is that the codes are only meant for spatial images of type I RDH. In [16] the code construction process proposed in [15] was generalized to extend the applicability to type II RDH. In [16] it is proved that rate-distortion bound can be reached in code construction; AAC (Adaptive Arithmetic Coder)'s decompression part is used as the embedding code. This paper contributes the following.

- Recognizes the efficiency of the approaches in [16] for binary covers and demonstrates that in prototype application.
- Develops recursive codes for gray scale covers and implements in the prototype.

The rest of the paper is organized in many sections. Section II presents review of literature. Section III presents the construction of recursive codes for both binary and gray scale covers. Section IV provides information on the experiments and evaluation while section V concludes the paper.

2. PRIOR WORK

2.1 Recursive Construction for Binary Covers

This section provides necessary details pertaining to recursive construction for binary covers.

2.2 Binary Covers

Binary cover is a digital image in which each pixel has only two possible values. In other words each pixel is represented by either 1 or 0. Generally two colors are associated with a binary image. They are black and white.

2.3 Recursive Construction

Kalker and Willems [17] proposed recursive construction which is a recursive embedding method where non reversible embedding code is used along with a conditional compression algorithm. Its general procedure is as described here. First of all a non reversible embedding code, embedding rate with given distortion is selected. The embedding process is done block by block sequentially. The first block in image represented by x_1 is subjected to message embedding and that result in y_1 which is a marked block. When the recipient knows y_1 , he can reconstruct the original cover image. This process continues recursively for all blocks. The recursive construction has advantages over traditional approach. However, it can't reach the upper bound as denoted in the following equation [16].

$$\text{Prev}(P_o, \Delta) = H_2(\max(P_o - \Delta, \frac{1}{2})) - H_2(P_o)$$

To overcome this drawback, the following sub section introduces improved recursive construction.

2.4 Improved Recursive Construction

This section improves recursive construction of codes for binary images in order to approach rate-distortion bound for any given distortion constraint. The process of the improved recursive function is illustrated in fig. 2.

Index	1	2	3	4	5	6	7	8	9	10	Second Block
message	0	1	0	1	1	1	0
y_1	0	0		0	0	0	1	0	0	1	
x_1	0	0	1	0	0	0	0	0	0	0	...
y_1	0	0	1	0	0	0	1	0	0	1	Comp(x_1)...
x_1'			1				0			0	
Comp(x_1')						Comp(x_1')					

Fig. 2 – Improved recursive construction [16]

As can be seen in fig. 2, message bits are presented. For each message bit index positions are given. Assuming that the embedding algorithm embeds messages only into zero symbols, the improved recursive construction is described here. x_1 represents the first cover block which has nine "0"s and one "1". As algorithm can embed only at zero symbols, 0.9 is the embedding rate while the distortion constraint is assumed to be 0.2. Therefore at zero symbols, distortion is 2/9. The decompression algorithm takes these as parameters. Assume that y_1 dash represents a 9 bit sequence which is derived from the 7 bits of message to be embedded. y_1 represents the first marked block that is derived by replacing "0"s in x_1 with y_1 dash. In y_1 , the index positions are {3, 7, 10}. According to this extract bits from x_1 and get x_1 dash which results in {1, 0, 0}. The x_1 dash then is compressed and embedded into the second block. This process continues recursively for all blocks.

3. STEGANOGRAPHY FOR GRAY SCALE COVERS

This section provides the proposed approach for reversible data hiding through steganography for gray scale covers.

3.1 Gray Scale Covers

Gray scale image is an image with a range of shades of gray. It has no specific colors. However, the darkest shade is considered black and lightest shade is considered white. In gray scale image generally each pixel is stored as byte. The value of such pixel is in the range 0 to 255.

3.2 Steganography

Steganography is a process of hiding text into images. This is one of the data hiding methods. The data hidden in image cannot be viewed by human eye. The process of embedding text into image is known as encoding while the process of extracting data from image is known as decoding. The compression techniques allow the process to work faster and efficiently. In this paper we used steganography with gray scale covers. Actually input is given as bitmap but the program converts it into gray scale as pre-processing and then performs encoding activity. The application we built supports both zip compression and arithmetic compression.

4. PROTOTYPE APPLICATION

The prototype application is built using Java platform. The environment used is 2 GM RAM with Core 2 Dual processor. Net Beans is used as Integrated Development Environment (IDE). Java's SWING API is used to build Graphical User Interface (GUI). The main UI of the application is as shown in fig. 3.

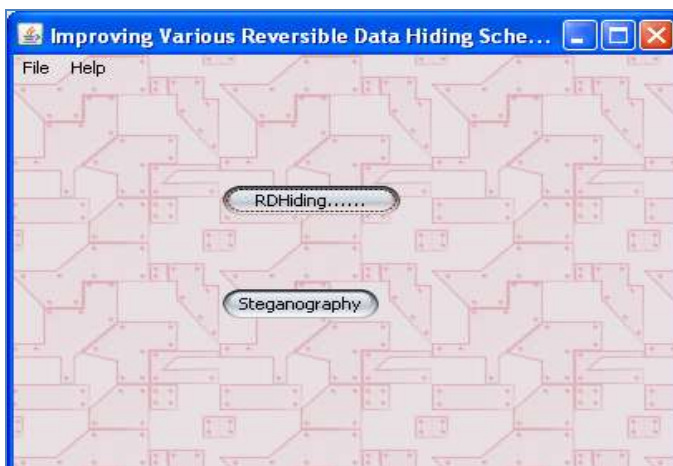


Fig. 3 – The main UI of the prototype

As can be seen in fig. 3, it is the main user interface for demonstrating the proof of concept with respect to both data hiding methods such as recursive codes for binary covers and steganography for gray scale covers. In either approaches the encoding decoding functionalities need similar user interface which is as shown in fig. 4 and fig. 5.

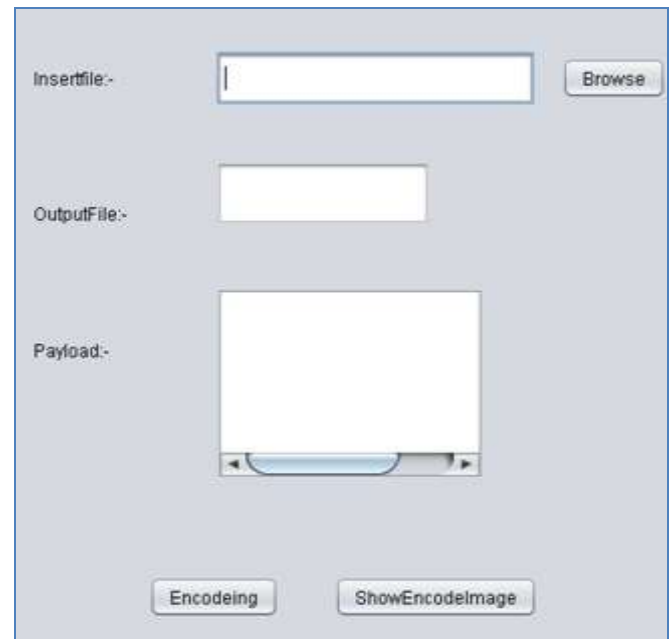


Fig. 4 – User Interface for Encoding

As can be seen in fig. 4 the encoding process takes three inputs. They are the image file, output file path, and payload. After encoding, it can show the encoded image when “Show Encoded Image” button is clicked.

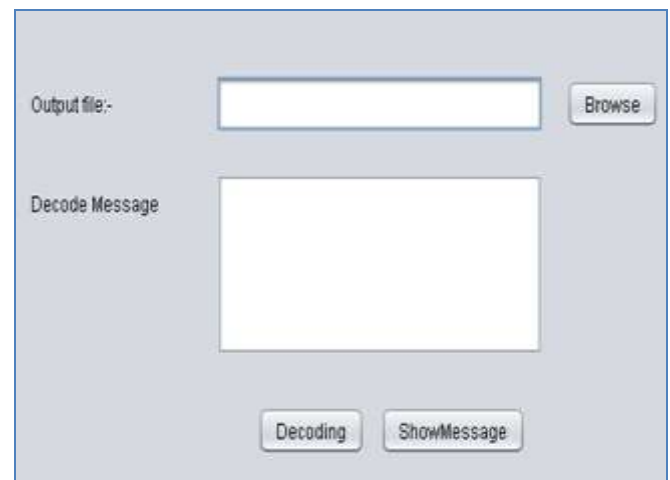


Fig. 5 – User Interface for Decoding

As can be seen in fig. 5 the decoding process takes two inputs. They are the stego file and the decoded message. After decoding, it can show the extracted message when “Show Message” button is clicked.

5. PERFORMANCE COMPARISON

This section provides experimental results and comparison of the proposed approach with prior approaches. In fact we developed a prototype application in Java to demonstrate the effectiveness of the proposed method. The two methods were compared using the embedding rate set to 0.7 and 0.9 while color blocks size is set to 200. The experiments are made with various image datasets of type gray scale and binary.

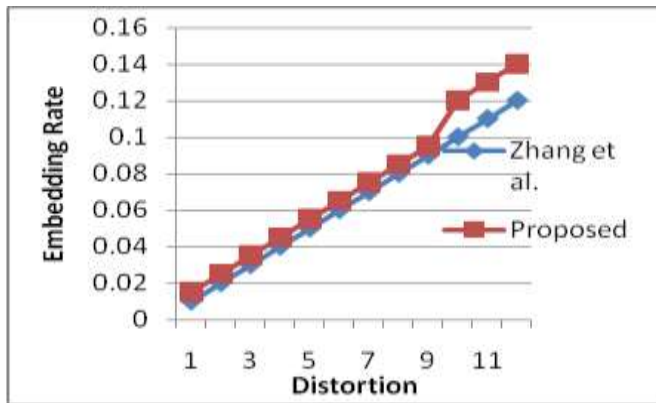


Fig. 6 – Comparison of Embedding Rate vs. Distortion

As can be seen in fig. 6 (a), it is evident that the proposed codes less distortion and high embedding rate. The “0” pixels rate is considered 0.7 and 0.9 respectively.

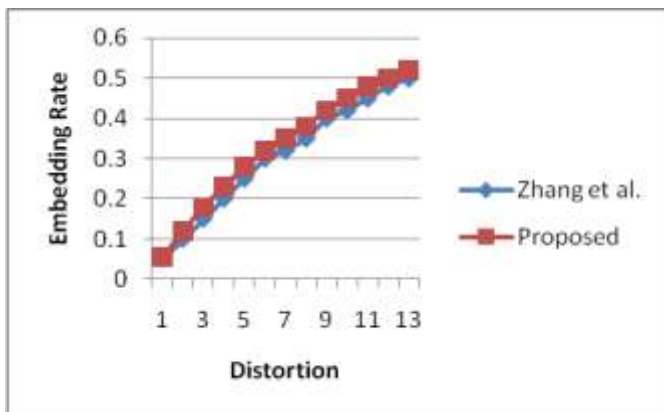


Fig. 7 – Comparison of Embedding Rate vs. Distortion

As can be seen in fig. 7, it is evident that the proposed codes less distortion and high embedding rate. The “0” pixels rate is considered 0.7 and 0.9 respectively.

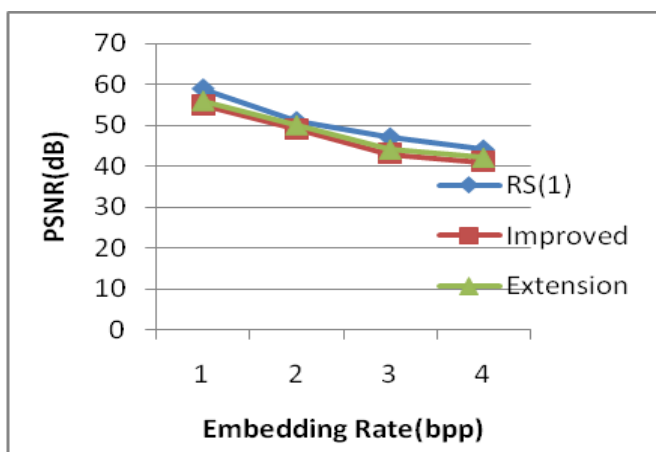


Fig. 8–Performance with Lenna Dataset

As can be seen in fig. 8, the vertical axis represents PSNR (Peak Signal-to-Noise Ratio) while the horizontal axis represents the embedding rate which is measured in bits carried by per pixel (bpp). The performance is shown for Lenna images.

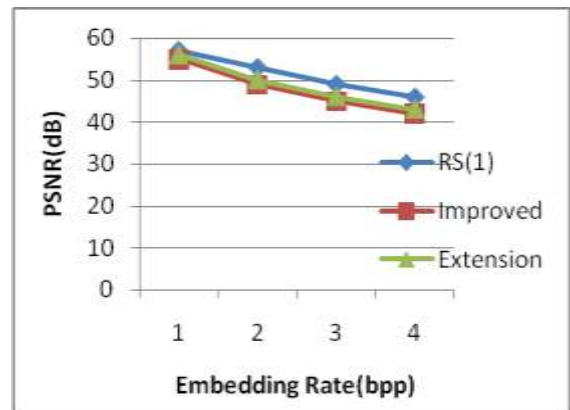


Fig. 9 – Performance with Baboom Dataset

As can be seen in fig. 9, the vertical axis represents PSNR (Peak Signal-to-Noise Ratio) while the horizontal axis represents the embedding rate which is measured in bits carried by per pixel (bpp). The performance is shown for Baboom images.

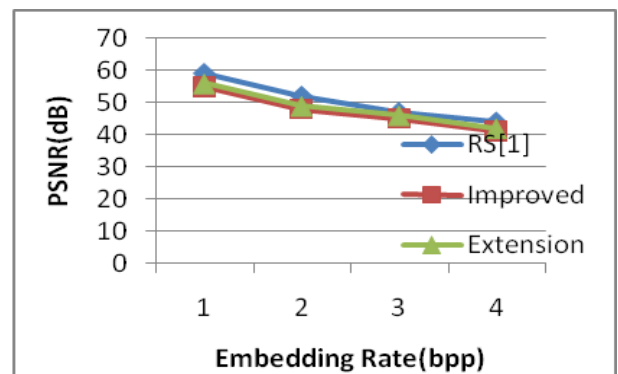


Fig. 10–Performance with Boat Dataset

As can be seen in fig. 10, the vertical axis represents PSNR (Peak Signal-to-Noise Ratio) while the horizontal axis represents the embedding rate which is measured in bits carried by per pixel (bpp). The performance is shown for Boat images.

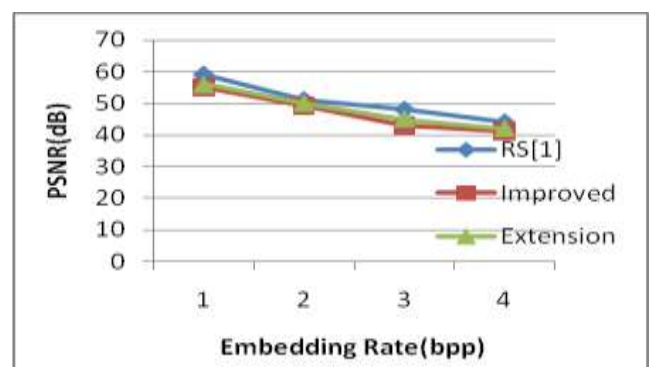


Fig. 11 – Performance with Barbara Dataset

As can be seen in fig. 11, the vertical axis represents PSNR (Peak Signal-to-Noise Ratio) while the horizontal axis represents the embedding rate which is measured in bits carried by per pixel (bpp). The performance is shown for Barbara images.

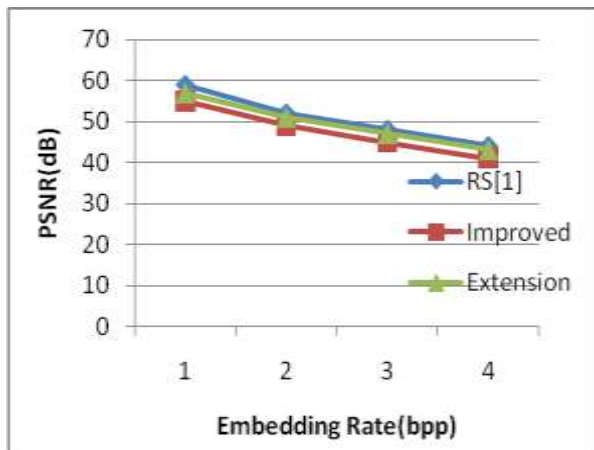


Fig. 12 – Performance with Goldrill Dataset

As can be seen in fig. 12, the vertical axis represents PSNR (Peak Signal-to-Noise Ratio) while the horizontal axis represents the embedding rate which is measured in bits carried by per pixel (bpp). The performance is shown for Goldrill images.

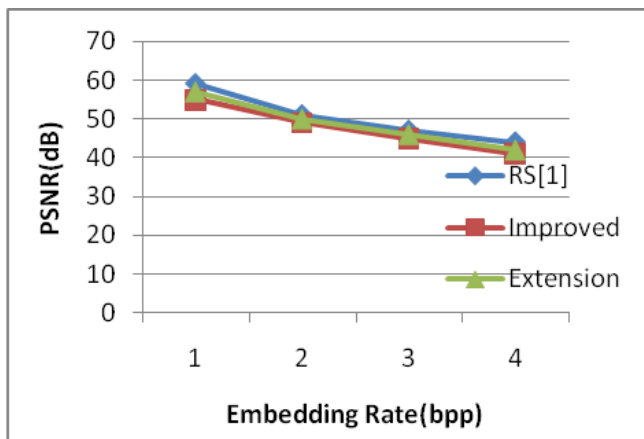


Fig. 13 – Performance with Peppers Dataset

As can be seen in fig. 13, the vertical axis represents PSNR (Peak Signal-to-Noise Ratio) while the horizontal axis represents the embedding rate which is measured in bits carried by per pixel (bpp). The performance is shown for Peppers images.

6. CONCLUSION

Many RDH methods perform message embedding and feature extraction separately. However, according to Kalker and Willems [17] it is possible to achieve high embedding rate when those are combined. In [16] this idea is followed using recursive codes for binary covers. This has outperformed the previous codes [17], [18]. This paper explores [16] besides developing recursive codes for binary covers by building a prototype application in Java platform. The prototype also demonstrates steganography for gray scale covers. The experimental results with various image datasets of both gray scale and binary revealed that the proposed methods are effective in RDH.

REFERENCES

- [1]. J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in *Proc. EI SPIE, Security Watermarking Multimedia Contents IV*, San Jose, CA, 2002, vol. 4675, pp. 572–583.
- [2]. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.
- [3]. S. Li and A. C. Kot, "Privacy protection of fingerprint database using lossless data hiding," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2010, pp. 1293–1298.
- [4]. Y.-A. Ho, Y.-K. Chan, H.-C. Wu, and Y.-P. Chu, "High-capacity reversible data hiding in binary images using pattern substitution," *Comput. Standards Interfaces*, vol. 31, no. 4, pp. 787–794, Jun. 2009.
- [5]. R. Du and J. Fridrich, "Lossless authentication of MPEG-2 video," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 2, pp. II-893–II-896.
- [6]. K. Wong, K. Tanaka, K. Takagi, and Y. Nakajima, "Complete video quality-preserving data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 10, pp. 1499–1512, Oct. 2009.
- [7]. J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
- [8]. A. M. Alattar, "Reversible watermark using difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.
- [9]. 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. Forensic Security*, vol. 3, no. 3, pp. 456–465, Sep. 2008.
- [10]. D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.
- [11]. 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.
- [12]. Z. Ni, Y. Q. Shi, N. Ansari, and S. Wei, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
- [13]. P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, no. 6, pp. 1129–1143, Jun. 2009.
- [14]. L. X. Luo, Z. Y. 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.
- [15]. W. Zhang, B. Chen, and N. Yu, "Capacity-approaching codes for reversible data hiding," in *Proc 13th IH*, 2011, vol. 6958, LNCS, pp. 255–269, Springer-Verlag.
- [16]. Weiming Zhang, Biao Chen, and Nenghai Yu, "Improving Various Reversible Data Hiding Schemes Via Optimal Codes for Binary Covers," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, VOL. 21, NO. 6, JUNE 2012.

[17]. T. Kalker and F.M.Willems, "Capacity bounds and code constructions for reversible data-hiding," in *Proc. 14th Int. Conf. DSP*, 2002, pp.71–76.

[18]. D. Maas, T. Kalker, and F.M.Willems, "A code construction for recursive reversible data-hiding," in *Proc. Multimedia Security Workshop ACM Multimedia*, Juan-les-Pins, France, Dec. 6, 2002.

BIOGRAPHIES



B.S.H. Shayeez Ahamed, I had done my Master's from the Jawaharlal Nehru Technological University, Anantapur, Andhra Pradesh. For the past 2 years my research work is based on the development of Embedding Images in Image Processing.

I am very proud to publish this journal for the reference to others.



Kumar G.J., I had done M.S from the University of Bedfordshire, United Kingdom at 2013. For the past 2 years my research work is based on the development of my own product in the domain of Image Processing and Data Schemas. I'm quite

happy with the data and would like to present it and write it up as a research paper.