# High Capacity Data Embedding in Images by Pixel Triplets Matching

R Sunder
Research Scholar
Manonmaniam Sundaranar University
Tirunelveli

P Eswaran
Assistant Professor
Alagappa University
Karaikudi

A Nagalinga Rajan
Research Scholar
Manonmaniam Sundaranar University
Tirunelveli

S Poonkuntran
Assistant Professor
Velammal College of Engineering and Technology
Madurai

## ABSTRACT

In this paper a data hiding method based on pixel pair matching (PPM) is presented. Pixel pair matching uses a pair of pixel intensity values as reference coordinates and searches the coordinate neighborhood according to the given message digit. The searched coordinate which encodes the message digit is put in place of the pixel pair. Exploiting modification direction (EMD), Diamond Encoding (DE) and Adaptive Pixel Pair Matching (APPM) are recently proposed methods based on PPM. Among these Adaptive Pixel Pair Matching provides the highest capacity of embedding with a given distortion. This paper presents an extension of the method to higher dimensional coordinate system and in particular with Pixel Triplet Matching (PTM). The experiments showed that PTM achieves higher capacity while being computationally feasible. The method is also resistant to simple steganalysis with adjacent HCF-COM.

## General Terms

Information hiding, Steganography

## Keywords

Steganography, Diamond Encoding, Least Significant Bit, Pixel Pair Matching.

## 1. INTRODUCTION

Steganography is a technique in which a secret message is embedded in a carrier medium and transmitted to the intended receiver [1]. In addition to the message being hidden from any possible intercepting agent there is the added advantage of the communication not seen as surreptitious and suspected by others. Digital Images provide a high capacity medium and is widespread in the internet thereby offering an ideal steganographic medium. A good data hiding method must be able to hide a relatively large amount of information measured in Bits per Pixel (BPP) and cause minimum distortion of the carrier image. Distortion is usually measured by the Mean Squared Error (MSE) and should be small enough to avoid detection by visual and statistical analysis.

Least Significant Bit (LSB) of pixels offers the simplest way to hide data in images. LSB Replacement [2] [23-39] is the simplest method of Image steganography and is computationally the cheapest. But it leaves a characteristic mark on the image; specifically it increases the even valued pixels and decreases the odd valued ones. This asymmetry can be reliably detected by statistical analysis methods [3]. An improvement over LSB Replacement came in the form of LSB Matching in which the pixel whose LSB do not match the message bit is either incremented or decremented by one based on a random choice. A simple and efficient modification was proposed by Chan et al to adjust the non-LSB bits towards least distortion known as Optimal Pixel Adjustment Process (OPAP).

In 2006, Mielikainen [4] proposed an embedding method in which a pair of pixels is taken as the embedding unit. Mielikainen’s method offers an MSE of 0.375 for the capacity of 1 bpp which is an improvement over LSB Replacement method’s MSE of 0.5. This method is a special case of a more general family of methods called Pixel Pair Matching (PPM). Zhang and Wang [5] proposed Exploiting Modification Direction (EMD) based on PPM, which offers an embedding capacity of 1.161 bpp. In 2009, Chao et al presented the Diamond Encoding (DE) [6] method. In DE the pixel pair is used as a coordinate in a reference grid. Each coordinate is assigned a Diamond Characteristic Value (DCV). The neighborhood of the pixel pair in the cover image is searched to find the coordinate which matched the message digit. The original pixel pair is then replaced with the searched coordinate. DE offers high capacity embedding with controllable distortion. In 2012, Hong and Chen proposed Adaptive Pixel Pair Matching [7] (APPM) which reduces the distortion of DE by optimizing the neighborhood set and the DCV function. Additionally any base can be used to encode the message.

Here an extension of the APPM method to use triplets of pixel values as coordinates and search in a three-dimensional neighborhood set is presented. The method allows the message to be encoded in a higher base thereby providing high capacity. It is also shown to be computationally feasible.

The rest of the paper is organized as follows. Section 2 reviews the related methods in brief; Section 3 presents the extension of APPM to three dimensions. Experimental results are presented in Section 4. Section 5 provides concluding remarks.

## 2. REVIEW OF STEGANOGRAPHIC METHODS

### 2.1 LSB Embedding

The Least Significant Bit is the obvious choice for a carrier stream within an Image. Any change in it causes the least distortion. For security purposes the pixels in the image are visited in a random order represented by a shared key and the LSB is replaced with the next message bit to be embedded. In this method even pixel values are always increased and odd values decreased creating a detectable statistical pattern. To avoid this problem LSB Matching changes the pixel values by adding either +1 or -1 to match the message bit. The choice is completely random and the asymmetry is avoided. In both these methods the carrying capacity is limited to 1 bpp with an expected MSE of 0.5.

## 2.2 Optimal Pixel Adjustment Process (OPAP)

OPAP was introduced in 2004 by Chan et al. It ameliorated the image distortion problem in LSB Embedding. In OPAP while embedding r message bits whose decimal value is s into a pixel v there are three choices for the modified pixel namely v' which is the LSB replacement result, and . The choice which yields the lowest distortion while remaining within the bounds of image representation namely 0 and 255 is taken as the modified pixel value and this value replaces the original pixel. All three choices contain the message bits in the r LSB bits and can be extracted easily.

## 2.3 Diamond Encoding (DE)

In 2009 Chao et al presented diamond encoding based on PPM. The secret message is encoded in B-ary notation into a pair of pixels. Here B is given by

$$B = 2k^2 + 2k + 1, k \geq 1$$

When k=1 DE reduces to EMD which conceals a message in 5-ary notation. The capacity of DE is (1/2)log2(2k2+2k+1) bpp.

Let the image dimensions be M×N and the secret message be S. A suitable k is chosen such that base of the notation B allows the message to be embedded in the cover image. More specifically B and hence k are chosen such that

$$\left|\frac{MN}{2}\right| \geq |S_B|$$

Here SB is the message in B-ary notation. The neighborhood set searched is a diamond shaped neighborhood with the original pixel pair as the center and a maximum radius of k. The neighborhood is formally given by

$$\Phi(x, y) = \big((a, b) \| a - x| + |b - y| \leq k\big)$$

Here x and y are the original pixel pair about to be replaced. The parameter k limits the distortion to the user defined levels. Each of the possible coordinates are assigned values in the range [0,B-1] by the diamond function given by f(x,y) = ((2k+1)x+y) mod B. The neighborhood is searched for a coordinate (x',y') whose DCV is equal to the message digit to be embedded. The original image pixels x and y are replaced with x' and y'. This procedure is repeated with the rest of the image pixel pairs till the entire message is embedded. In the extraction stage the image pixels are scanned in the same order and the message digits are extracted as the DCV of the pairs.

## 2.4 Adaptive Pixel Pair Matching

In 2012, Hong and Chen extended the basic idea of DE to include all possible neighborhood sets allowing the use of an arbitrary base. It is optimal in sense of minimum distortion. Specifically the neighborhood set and the DCV function are selected based on the optimization of the distortion. The optimization problem is expressed as the following.

$$\begin{gathered} Minimize \sum_{i=0}^{B-1} (x_i - x)^2 + (y_i - y)^2 \\ subject\ to\ the\ constraints \\ f(x_i, y_i) \in \{0,1, \dots, B-1\} \\ f(x_i, y_i) \neq f(x_j, y_j)\ if\ i \neq j \\ for\ 0 \leq i, j \leq B-1 \end{gathered}$$

Here f, the DCV function is given by

$$f(x, y) = (x + c_B y)\ modulo\ B$$

For a given value of B, the $(x_i,y_i)$ values which represent the neighborhood set Φ and the constant CB can be found by solving (1). Some of the results for B = 4, 16 and 25 are shown in Figure 1.

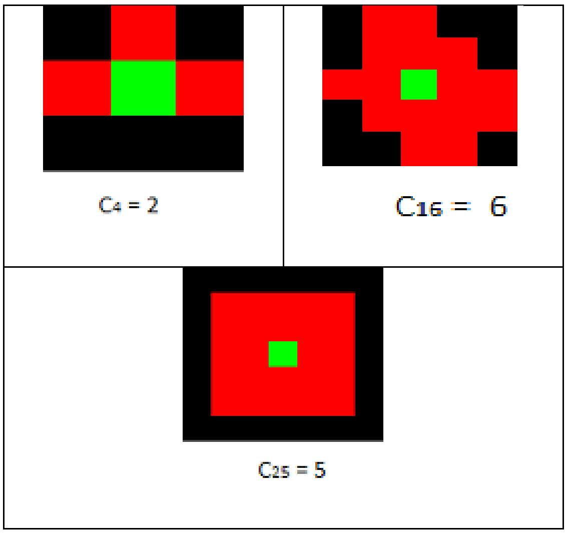

**Figure 1 APPM Values**

## 3. EXTENSION OF APPM TO PIXEL TRIPLETS

In this paper an extension of the notion of pixel pair matching to pixel triplets is presented. This section presents a formulation of the optimization in three dimensional coordinates, derive solutions, and give the algorithm for embedding and extraction of messages.

Let the pixel triplet be $(x_i, y_i, z_i)$ and the message digit be m which is in B-ary notation. The base B depends on the distortion radius k and is given by

$$B(k) = \left(\frac{4}{3}\right)k^3 + 2k^2 + \left(\frac{8}{3}\right)k + 1$$

In contrast to this the DE and APPM method allows the use of Base given by

$$B(k) = 2k^2 + 2k + 1$$

Compared to DE and APPM, using three pixels as coordinates allows us to use a larger base and hence a larger message for a given distortion.

Now the optimization problem may be posed as finding the closest coordinate with the DCV equal to m. The DCV function in three dimensions may be formulated as

$$f(x, y, z) = (c_1 x + c_2 y + c_3 z)\ modulo\ B$$

Here $c_1$, $c_2$, $c_3$ are constants that needs to be determined. The Neighborhood Φ and the constants are determined based on the following optimization criteria.

$$Minimize \sum_{i=0}^{B-1} ((x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2)$$

$$f(x_i, y_i, z_i) \in \{0,1, \ldots, B-1\}$$

$$f(x_i, y_i, z_i) \neq f(x_j, y_j, z_j)\ for\ i \neq j$$

$$0 \leq i, j \leq B - 1$$

The constants are calculated by a simple parameter sweep and since this is a one-time calculation the computational efficiency is not of undue importance. The i, j values which represent the neighborhood lattice is easily calculated by solving the above optimization problem.

The different bases for their corresponding distortion is shown compared to that of the DE, APPM methods in Table 2. The list of constants is given in Table 3. The neighborhood lattices are shown diagrammatically in Table 4.

**Table 1 Base for Different Distortions**

| Distortion *k* | Base (DE) | Base (Triples) |
|---|---|---|
| 1 | 5 | 7 |
| 2 | 13 | 25 |
| 3 | 25 | 63 |
| 4 | 41 | 129 |
| 5 | 61 | 231 |
| 6 | 85 | 377 |
| 7 | 113 | 575 |
| 8 | 145 | 833 |

It can be seen from Table 1 that using triplets enables us to pack more message bits for the approximately same distortion. The visualization of the lattices shows that they are as densely packed as possible for the given base.

**Table 2 List of Constants for different bases**

| *B* | $C_1$ | $C_2$ | $C_3$ | *B* | $C_1$ | $C_2$ | $C_3$ | *B* | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **2** | **1** | **1** | **1** | **23** | **1** | **3** | **8** | **44** | **1** | **14** | **20** |
| **3** | **1** | **1** | **1** | **24** | **1** | **3** | **8** | **45** | **1** | **4** | **17** |
| **4** | **1** | **1** | **2** | **25** | **8** | **11** | **12** | **46** | **1** | **8** | **12** |
| **5** | **1** | **1** | **2** | **26** | **1** | **3** | **9** | **47** | **1** | **4** | **18** |
| **6** | **1** | **2** | **3** | **27** | **3** | **4** | **9** | **48** | **1** | **4** | **18** |
| **7** | **1** | **2** | **3** | **28** | **1** | **3** | **9** | **49** | **6** | **7** | **17** |
| **8** | **1** | **2** | **3** | **29** | **1** | **10** | **13** | **50** | **1** | **8** | **12** |
| **9** | **1** | **2** | **3** | **30** | **3** | **5** | **9** | **51** | **3** | **5** | **21** |
| **10** | **1** | **2** | **3** | **31** | **1** | **11** | **14** | **52** | **2** | **10** | **13** |
| **11** | **1** | **2** | **3** | **32** | **4** | **6** | **9** | **53** | **1** | **4** | **21** |
| **12** | **1** | **2** | **4** | **33** | **3** | **5** | **9** | **54** | **1** | **4** | **20** |
| **13** | **1** | **2** | **4** | **34** | **2** | **3** | **11** | **55** | **1** | **5** | **21** |
| **14** | **1** | **2** | **5** | **35** | **1** | **11** | **16** | **56** | **2** | **3** | **22** |
| **15** | **1** | **2** | **5** | **36** | **6** | **8** | **9** | **57** | **1** | **5** | **22** |
| **16** | **1** | **2** | **6** | **37** | **1** | **6** | **9** | **58** | **1** | **5** | **22** |
| **17** | **1** | **2** | **6** | **38** | **1** | **6** | **9** | **59** | **1** | **7** | **24** |
| **18** | **1** | **2** | **6** | **39** | **1** | **12** | **18** | **60** | **2** | **7** | **26** |
| **19** | **1** | **2** | **6** | **40** | **4** | **14** | **19** | **61** | **23** | **25** | **30** |
| **20** | **2** | **5** | **6** | **41** | **1** | **5** | **13** | **62** | **2** | **7** | **27** |
| **21** | **2** | **5** | **6** | **42** | **3** | **8** | **12** | **63** | **1** | **5** | **25** |
| **22** | **1** | **3** | **8** | **43** | **1** | **4** | **15** | **64** | **1** | **9** | **30** |

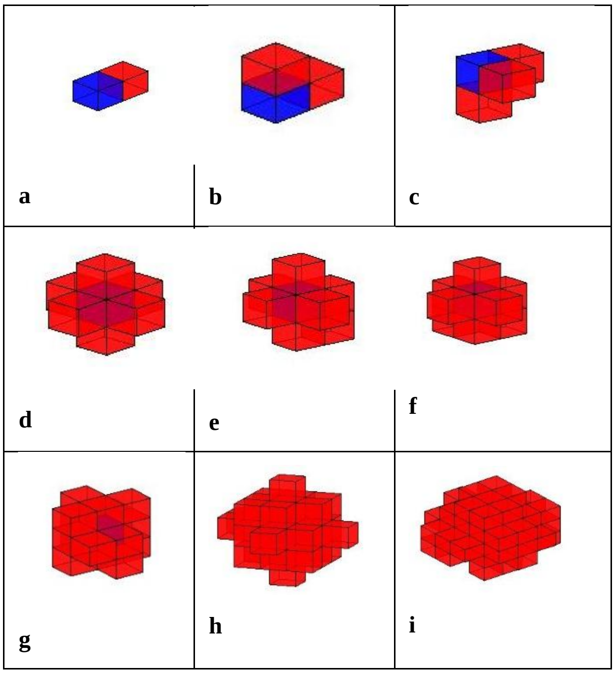

**Figure 2 Visualization of Neighborhood sets for Base a) 2 b) 3 c) 4 d) 7 e) 8 f) 9 g) 16 h) 32 i) 64.**

The algorithms for embedding and extraction are given as follows.

### 3.1 Embedding Procedure

Input: Cover Image I of size M×N, secret bit stream S, and Shared Key K.

Output: Stego Image I', $C_1$, $C_2$, $C_3$, $\Phi_B$ and K

1. Find the minimum B that satisfies $\left\lfloor \frac{M \times N}{3} \right\rfloor \geq S_B$
2. Solve the discrete optimization problem to find $C_1$, $C_2$, $C_3$ and $\Phi_B$
3. Encode the secret bit stream S in B-ary notation $S_B$.
4. Construct a pseudorandom non-repeat sequence Q with K as the seed.
5. Take three pixels from the cover image I in the sequence given by Q and find the coordinate in the neighborhood $\Phi_B$ with these three pixels as the center that has a DCV equal to the next message digit from $S_B$
6. Replace the three pixels with the searched coordinates.
7. Repeat steps 5 and 6 till all the message digits $S_B$ are embedded.

Consider as an example a pixel triplet of (5,110,120) with the base selected as 16 and the message digit as 11. The DCV function is given aS

$$f(x, y, z) = (x + 2y + 6z)\ modulo\ 16$$

8. With this as the DCV function, the neighborhood lattice with center at (5,110,120) is searched for a coordinate with DCV of 11 which is found as (5, 110, 119). The pixel values are replaced with these new values.
9. The constants and the lattices need not be solved every time but can be computed once and stored for further use. These are assumed to be available at both the sender and receiver sides.

### 3.2 Extraction Procedure

The message can be extracted by finding the DCV of the triples of pixels in the same order which can be generated with the knowledge of shared key K.

Input: Stego Image I', $C_1$, $C_2$, $C_3$, $\Phi_B$ and K

Output: The Embedded Message S.

1. Construct the embedding sequence Q using the key K as seed for the pseudorandom order generator.
2. Select three pixels (x', y', z') according to the sequence Q.
3. Calculate f (x', y', z') which is the embedded digit.
4. Repeat steps 2 and 3 till all the message digits are extracted.
5. The message stream S is obtained by converting the message digits into binary representation.

Continuing the example the pixels which are (5,110,119) are read and their DCV is calculated as 11 which is the embedded message digit. The distortion in the case is a squared error of 1.

## 4. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the results of experiments that compare the above discussed Adaptive Pixel Triplet Matching (APTM) with APPM. The test images used are shown Figure 3.

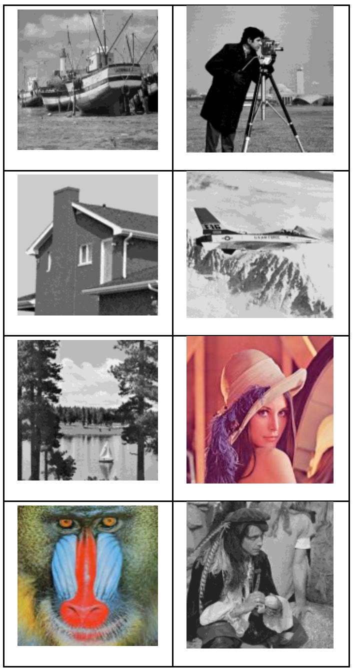

**Figure 3 Test Images**

The Embedding and Extraction algorithms are implemented in MATLAB without much optimization and take 2-3 seconds each in a workstation with Core i3 processor and 4GB RAM. To find the distortion, Peak Signal to Noise Ratio (PSNR) is used. Given two images I1 and I2 the Mean Squared Error (MSE) between them is defined as

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(I_1(i,j) - I_2(i,j)\right)^2$$

Here M and N represent the size of the images. PSNR is then defined on a logarithmic scale in decibels.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE}\right)$$

The Figure 4 show the variation of PSNR with respect to the capacity of embedding measured in Bits per pixel (bpp). Again APPM and APTM are shown for comparison. The improvement in PSNR is clearly seen in higher capacities.

Figure 5 shows the comparison of using different bases for the same capacity in APTM. Naturally using the lowest possible base as dictated by the message size yields the best results.

The results of experiments are shown in Figure 6. For each image APPM and APTM are applied with the same message with different Bases. The improvement in PSNR is clearly seen. For a given message using triples improves the PSNR by embedding with smaller value of across distance of the neighborhood. The improvement is considerably pronounced when larger bases are used as expected.

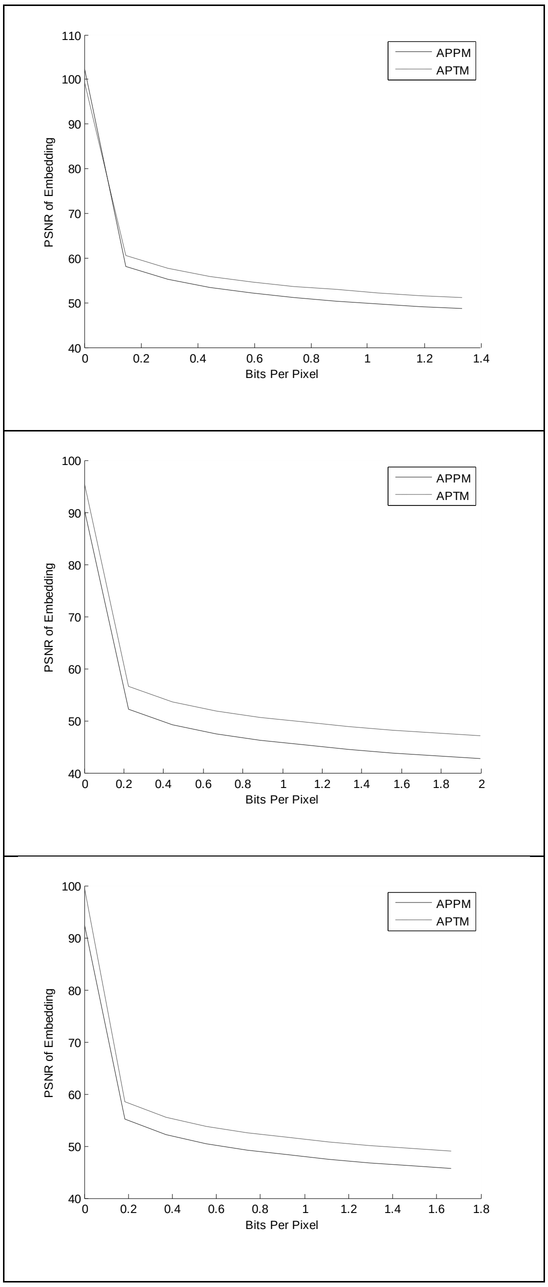

**Figure 4 PSNR of Embedding vs. BPP for three test images**

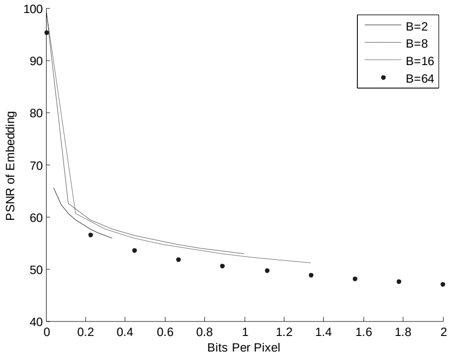

**Figure 5 PSNR vs. BPP for APTM**

Any steganography technique must also withstand steganalysis so that it cannot be reliably detected. To this end the results of adjacent HCF-COM detector introduced by A D Ker on the APTM method is presented. In the Receiver Operating Characteristic (ROC) curve the Area Under Curve (AUC) is a measure of the reliability of the detector. As shown in the ROC curves the method cannot be detected reliably and does not leave a detectable pattern on the adjacency histogram.

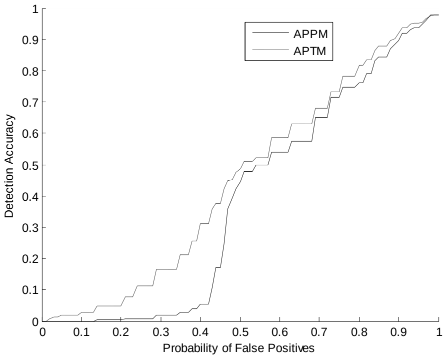

**Figure 6 ROC Curves for ADJ-HCF-COM**

**Table 3 Comparison of APPM with APTM**

| Image | Adaptive Pixel Pair Matching | | | | Adaptive Pixel Triple Matching | | | | Improvement in PSNR | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B =2 | B = 8 | B = 16 | B = 64 | B =2 | B = 8 | B = 16 | B = 64 | B = 2 | B = 8 | B = 16 | B = 64 |
| Boat | 55.90 | 51.92 | 48.62 | 42.71 | 55.89 | 52.90 | 51.16 | 47.11 | -0.01 | 0.98 | 2.54 | 4.40 |
| Cameraman | 55.93 | 51.94 | 48.61 | 42.72 | 55.88 | 52.91 | 51.14 | 47.11 | -0.05 | 0.96 | 2.53 | 4.39 |
| House | 55.60 | 51.89 | 48.63 | 42.73 | 56.00 | 52.91 | 51.15 | 47.11 | 0.39 | 1.02 | 2.52 | 4.37 |
| Jet plane | 55.87 | 51.92 | 48.59 | 42.73 | 55.90 | 52.90 | 51.14 | 47.11 | 0.03 | 0.97 | 2.55 | 4.38 |
| Lake | 55.91 | 51.94 | 48.60 | 42.74 | 55.93 | 52.89 | 51.14 | 47.11 | 0.02 | 0.95 | 2.54 | 4.37 |
| Lena (color) | 55.91 | 51.93 | 48.61 | 42.73 | 55.92 | 52.89 | 51.12 | 47.11 | 0.01 | 0.96 | 2.51 | 4.38 |
| Mandrill (color) | 55.92 | 51.93 | 48.61 | 42.73 | 55.90 | 52.90 | 51.13 | 47.10 | -0.02 | 0.97 | 2.53 | 4.38 |
| Pirate | 55.89 | 51.93 | 48.61 | 42.73 | 55.92 | 52.89 | 51.14 | 47.10 | 0.03 | 0.96 | 2.53 | 4.37 |

## 5. CONCLUSION AND FUTURE WORK

An extension of pixel pair matching to three pixels is presented. The APTM method can provide a larger embedding capacity with a given distortion by allowing the message to be encoded in a larger base notation. The method is also computationally feasible and can be used in practical situations. It is also resilient to steganalysis technique. Extending the idea further into higher tuples of pixels is straightforward but is only limited by computational complexity of its implementation. Future work includes a thorough security analysis against state-of-art methods and extending the method to higher dimensions.

## 6. REFERENCES

[1] Ingemar. J. Cox et al, “Digital Watermarking and Steganography,”, 2nd ed. Morgan Kaufmann series in computer security.

[2] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[3] C. K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” Pattern Recognition., vol. 37, no. 3, pp. 469–474, 2004.

[4] J. Fridrich, M.Goljan, and R.Du, “Reliable detection of LSB steganography in color and grayscale images,” in Proc. Int. Workshop on Multimedia and Security, 2001, pp. 27–30.

[5] J. Mielikainen, “LSB matching revisited,” IEEE Signal Process. Lett., vol. 13, no. 5, pp. 285–287, May 2006.

[6] X. Zhang and S. Wang, “Efficient steganographic embedding by exploiting modification direction,” IEEE Commun. Lett., vol. 10, no. 11, pp. 781–783, Nov. 2006.

[7] R.M. Chao, H. C. Wu, C. C. Lee, and Y. P. Chu, “A novel image data hiding scheme with diamond encoding,” EURASIP J. Inf. Security, vol.2009, 2009, DOI: 10.1155/2009/658047, Article ID 658047.

[8] Wien Hong; Tung-Shou Chen; , "A Novel Data Embedding Method Using Adaptive Pixel Pair Matching," Information Forensics and Security, IEEE Transactions on , vol.7, no.1, pp.176-184, Feb. 2012 doi: 10.1109/TIFS.2011.2155062

[9] T. Filler, J. Judas, and J. Fridrich, “Minimizing embedding impact in steganography using trellis-coded quantization,” in Proc. SPIE, Media Forensics and Security, 2010, vol. 7541, DOI: 10.1117/12.838002.

[10] S. Lyu and H. Farid, “Steganalysis using higher-order image statistics,” IEEE Trans. Inf. Forensics Security, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[11] N. Provos and P. Honeyman, “Hide and seek: An introduction to steganography,” IEEE Security Privacy, vol. 3, no. 3, pp. 32–44, May/Jun. 2003.

[12] A. D. Ker, “Steganalysis of LSB matching in grayscale images,” IEEE Signal Process. Lett., vol. 12, no. 6, pp. 441–444, Jun. 2005.