International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 6, August - 2012

Comparison of Reversible Data hiding Techniques for Digital images

D.Manga Ratnam, L.Padmalatha
M.Tech Student, Associate Professor
Gudlavalleru Engineering College

Abstract

In this paper three different Reversible Data hiding techniques based on histogram was implemented. The first technique is based on pixel difference which is improvement to histogram modification technique. In order to communicate with multiple peak points another technique called binary tree has been implemented. Considering Human Visual System Characteristics recent technique has been implemented which can effectively reduce the distortion caused by data embedding.

1. Introduction

Data Hiding is a term including a wide range of applications for embedding messages in content [8],[9].Hiding information destroys the host image even though the distortion introduced by hiding is imperceptible to the human visual system. There are, however, some sensitive images for which any embedding distortion of the image is intolerable. Consequently, reversible data hiding techniques are designed to solve the problem of lossless embedding of large messages in digital images so that after the embedded message is extracted, the image can be completely restored to its original state before embedding occurred.

Reversible data hiding techniques have also been proposed for various fields such as audio, MPEG-2 video, 3-D meshes, visible watermarking, SMVQ-based compressed domain, and the integer-to-integer wavelet domain. Novel histogram-based reversible data hiding technique was presented by Ni et al. in [5], in which the message is embedded into the histogram bin. They used peak and zero points to achieve low distortion, but with attendant of low capacity. Many Histogram modification techniques have been extended recently in [1],[3]. However, those techniques all suffer from issue represented by the need to communicate pairs of peak and zero points to recipients.

The histogram modification technique using pixel differences is used to increase hiding capacity. Binary tree structure is used to eliminate the requirement to communicate pairs of peak and zero points to the recipient. Histogram shifting technique is adopted to prevent overflow and underflow. Histogram modification based reversible data embedding algorithm considering the human visual system (HVS) is adopted to reduce the distortion caused in data embedding. Considering the concept of just noticeable difference (JND) [4], [2], the pixels in the smooth and edge regions are differently treated to reduce perceptual distortion.

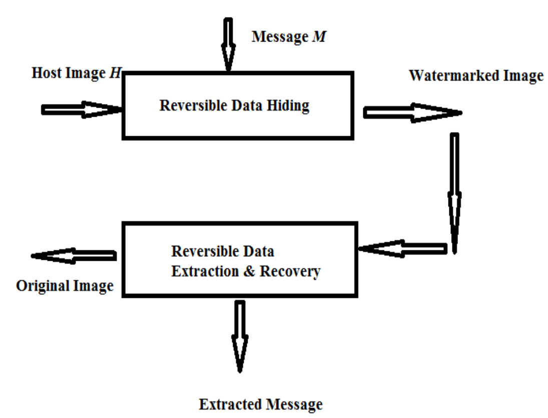

Fig. 1: Diagram for the reversible data hiding procedure

2. Histogram Modification Based on Pixel difference

In [3], Ni et al. introduced a reversible data hiding scheme based on histogram modification using pairs of peak and zero points. Let P be the value of peak point and Z be the value of zero point. The range of the histogram, P+1, Z-1, is shifted to the right-hand side by 1. Once a pixel with value P is encountered, if the message bit is "1," then pixel value is increased by 1. Otherwise, no modification is needed. Data extraction is actually the reverse of the data hiding process. The number of message bits that can be embedded into an image equals the number of pixels associated with the peak point.

However the histogram modification technique does not work well when an image has an equal histogram. While multiple pairs of peak and minimum points can be used for embedding, the pure payload is still a little low. Thus, an efficient extension of the histogram modification technique is considered in which the differences between adjacent pixels are taken instead of simple pixel value. Since image neighbour pixels are strongly correlated, the distribution of pixel difference has a prominent maximum. In addition, a binary tree structure is used to solve the issue about communicating multiple pairs of peak and minimum points to the recipient.

2.1. Algorithm

1. Scan the image H in an inverse s-order. Calculate the pixel difference d_i between pixels x_{i-1} and x_i by

$$d_i = \begin{cases} x_i, & if\ i = 0 \\ |x_{i-1} - x_i|, & otherwise \end{cases}$$

2. Determine the peak point P from the pixel differences.

3. Scan the whole image in the same inverse s-order as in Step 1. If $d_i > P$, shift xi by 1 unit

$$y_i = \begin{cases} x_i, & if\ i = 0\ or\ d_i < P, \\ x_i + 1, if & d_i > P\ and\ x_i \geq x_{i-1}, \\ x_i - 1, if & d_i > P\ and\ x_i < x_{i-1}, \end{cases}$$

Where y_i is the watermarked value of pixel i.

4. If $d_i = P$, modify x_i according to the message bit

$$y_i = \begin{cases} x_i + b, & if\ d_i = P\ and\ x_i \geq x_{i-1}, \\ x_i - b, & if\ d_i = P\ and\ x_i < x_{i-1}, \end{cases}$$

Where b is a message bit to be embedded.

5. At the receiving end, the recipient extracts message bits from the watermarked image by scanning the image in the same order as during the embedding. The message bit b can be extracted by

$$b = \begin{cases} 0, & if\ |y_i - x_{i-1}| = P \\ 1, & if\ |y_i - x_{i-1}| = P + 1 \end{cases}$$

Where x_{i-1} denotes the restored value of y_{i-1}.

6. The original pixel value of x_i can be restored by

$$x_i = \begin{cases} y_i + 1, if & |y_i - x_{i-1}| > P\ and\ y_i < x_{i-1}, \\ y_i - 1, if & |y_i - x_{i-1}| > P\ and\ y_i > x_{i-1}, \\ y_i, & otherwise. \end{cases}$$

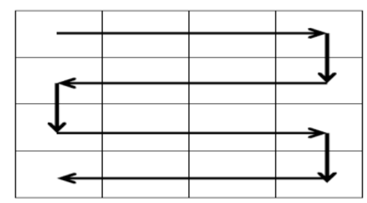

Fig. 2: Scan order: inverses-order

3. Binary Tree Structure

In a reversible data hiding scheme, large hiding capacities can be obtained by repeated data hiding processes. However, there have some problems with this scenario. First, recipients are not able to retrieve both the embedded message and the original host image without knowledge of peak points of every hiding pass. Even though this issue can be solved by supplying a side communication channel for these peak points, the fact that the amount of information that has to be communicated to the recipient through a side channel easily extends the embedded message length. Thus, we a binary tree structure is adopted to solve the problem of communication of multiple peak points.

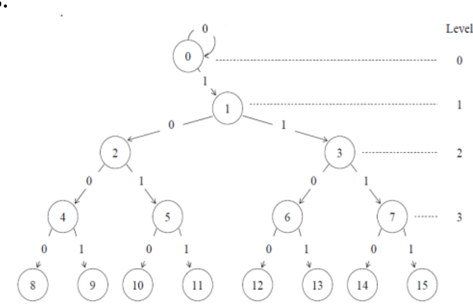

Fig.3: An auxiliary binary tree for the reversible data hiding Scheme

An auxiliary binary tree for solving the issue of communication of multiple peak points is given in Figure. Where each element denotes a peak point. Let us assume that the number of peak points that we use to embed messages is 2^L, where L is the level of the binary tree. Once a pixel difference d_i that satisfies $d_i < 2^L$ is encountered, if the message bit to be embedded is "0," the left child of the node d_i is visited; otherwise, the right child of the node d_i is visited. We note that higher payloads force us to use higher tree levels, thus quickly increasing the distortion in the image beyond an acceptable level. All the recipient need to share with the sender is the tree level L since an auxiliary binary tree is adopted that predetermines multiple peak points used to embed messages.

4. Human Visual System Characteristics

In this technique, a local causal window is used to predict a pixel value and estimate an edge. In case of video sequence motions blur and motion sharpening to be considered. Then, by considering the concept of just noticeable difference (JND) the pixels in the smooth and edge regions are differently treated to reduce perceptual distortion.

A pixel value is predicted using casual window as follows

$$\hat{x}(i,j)=\frac{1}{N(\Omega_{i,j})}\sum_{(m,n)\Omega_{i,j}} x(m,n)$$

Where $\Omega_{i,j}$, represents a causal window surrounding x(i,j) and $N(\Omega_{i,j})$ returns cardinality of the set $\Omega_{i,j}$ For instance, the causal window of size B=5 shown in Figure 4 contains 12 pixel positions and the average of the pixel values at these positions is used as a predicted value. Then we calculate the pixel difference between the original and predicted values by

$$d(i,j)=|\boldsymbol{x}(i,j)-\hat{x}(i,j)|$$

Where d (i.j) is the difference value used in the data embedding process.

The perceptual characteristic of the HVS is exploited to alleviate the quality degradation caused by data embedding. To this end, the edge is simply estimated for each pixel as follows:

$$E(i,j)=\begin{cases}1, if\ var(\Omega_{i,j}) > T_e\\ 0, if\ var(\Omega_{i,j}) \le T_e\end{cases}$$

Where E (i,j) indicates whether the pixel is the edge or not,var(Ωi,j) represents the variance of pixel values in Ωi,j and Te is an edge threshold.

(i-2,j-2)	(i-2,j-1)	(i-2,j)	(i-2,j+1)	(i-2,j+2)
(i-1,j-2)	(i-1,j-1)	(i-1,j)	(i-1,j+1)	(i-1,j+2)
(i,j-2)	(i,j-1)	(i,j)		

Fig.4: Casual window for computing E(i,j),jnd(i,j)

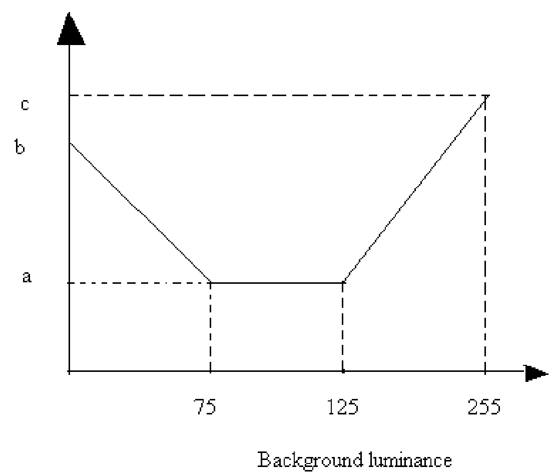

Fig.5: visibility threshold against background luminance

Since the HVS is known to perceive the difference above the JND, the JND value is estimated after edge detection as follows:

$$jnd(i,j)= T_l(i,j) + \lambda \frac{T_a(i,j)}{T_l(i,j)}$$

Where T_l and T_a are two thresholds representing the luminance adaptation and the activity masking of the HVS characteristics, respectively, and λ=0.5. In order to estimate T_l, background luminance is first measured by taking the average value of the local neighbourhood. Then, a piecewise linear approximation in Fig. 5 is used with three parameters, a, b and c, described earlier. Specifically, a=10, b=20, c=24 for non edge pixels and a=8, b=18, c=22 for edge pixels. In addition, Ta is defined as the maximum pixel difference value in the local neighbourhood. Note that when computing E(i,j) background luminance, T_l and T_a are, only the pixels in the causal window are used because only these pixels are available at the data extraction stage due to the raster scan order processing.

Actual data embedding is performed by increasing the difference value d(i,j) and finding the extra space that can contain to be embedded bits. Thus, the overflow and underflow problem can happen when the embedded value exceeds a pixel value bound (0 to 255 in 8 bit images). To solve this problem, the original image histogram is shrunk from both sides by 2^L , where L is the embedding level. To realize reversible data embedding, the overhead information describing this pre-processing is lossless compressed and embedded together with pure payload data.

Adjust the embedding level for each pixel according to the local image characteristics. For the nonedge pixel, the embedding level $K_{i,j}$ is defined by

$$K_{i,j} = arg_k \max 2^k \text{ , } subject\ to\ 2^k < \text{jnd}(i,j),\ k \leq L$$

A maximum possible embedding level is chosen with a constraint that the pixel value change should be lower than the JND value. This is because the distortion above the JND in the smooth region is perceptually disturbing.

On the other hand, for the edge pixel $K_{i,j}$ is determined by

$$K_{i,j} = arg_k \min 2^k \text{ , } subject\ to\ 2^k > \text{jnd}\ (i,j),\ k \leq L$$

A minimum possible embedding level above the JND is used to embed a sufficient amount of data. This is because it is difficult to find the extra space using the embedding level lower than the JND since the difference values in the edge region are high. Besides, the increase of the JND in the edge region does not severely deteriorate the visual quality and sometimes an intentional increase of the JND in the edge region is employed in the image enhancement algorithm.

After estimating the edge, the JND, and finally the embedding level, we can try to embed a message bit for each pixel. If d(i,j) < 2Ki,j the message bit is embedded by

$$y(i,j) = \begin{cases} x(i,j) + d(i,j) + b, x(i,j) \geq \hat{x}(i,j) \\ x(i,j) - d(i,j) - b, x(i,j) < \hat{x}(i,j) \end{cases}$$

Otherwise, if d(i,j) > $2^{K_{i,j}}$, data embedding is not performed but the difference value should be expanded to discriminate this pixel from the embedded pixels. In this case, the output pixel value is obtained by

$$y(i,j) = \begin{cases} x(i,j) + 2^{K_{i,j}},\ x(i,j) \geq \hat{x}(i,j) \\ x(i,j) - 2^{K_{i,j}},\ x(i,j) < \hat{x}(i,j) \end{cases}$$

At the data extractor,

If $|y(i,j) - \hat{x}(i,j)| < 2^{k_{i,j}+1}$, then message bit b is extracted by

$$b = \begin{cases} 0, \text{ If } |y(i,j) - \hat{x}(i,j)| \text{ is even} \\ 1. \text{ If } |y(i,j) - \hat{x}(i,j)| \text{ is odd} \end{cases}$$

When $|y(i,j) - \hat{x}(i,j)| < 2^{k_{i,j}+1}$, then pixel value is recovered by

$$x(i,j) = \begin{cases} y(i,j) + \left\lceil \frac{|y(i,j) - \hat{x}(i,j)|}{2} \right\rceil, \text{if } y(i,j) < \hat{x}(i,j) \\ y(i,j) - \left\lceil \frac{|y(i,j) - \hat{x}(i,j)|}{2} \right\rceil, \text{if } y(i,j) \geq \hat{x}(i,j) \end{cases}$$

5. Experimental Observations

In order to evaluate the performance of the proposed algorithm, different gray scale images are used. First, the capacity versus distortion performance of the above algorithms is illustrated.

For all test images, more bits can be embedded by increasing the embedding level at the expense of the quality of degradation. Since data embedding is dependent on the redundancy in the image content, images containing a large smooth area such as Candy can embed a large number of bits, whereas images with complicated textures such as Baboon can contain a relatively small number of bits.

5.1. Results

The performance comparison of different gray scale images plot was given below

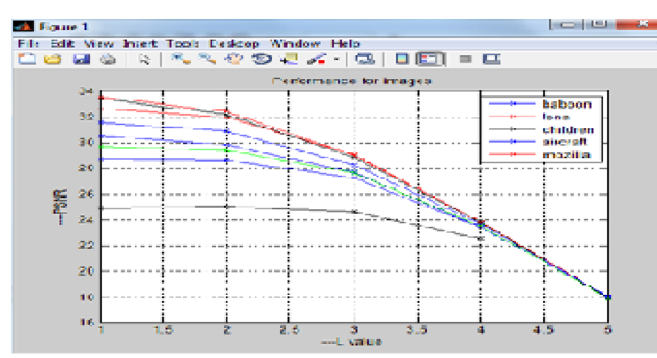

Fig.6: Plot of PSNR Vs L value for different images

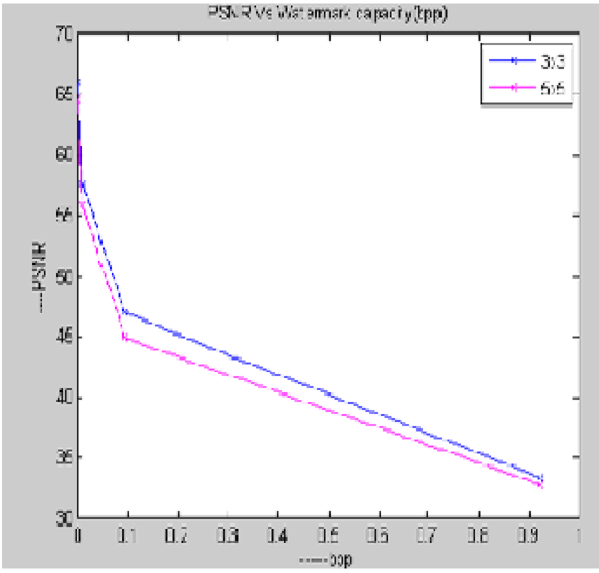

Fig.7: Plot of PSNR Vs bpp

From the graphs it was observed that PSNR value obtained from binary tree and pixel difference was very low of order 25-30 db in range where as PSNR value obtained from human visual system characteristics is 64db.Thus by using this HVS we can increase the signal strength.

7. Conclusion

For achieving copyright protection, robust digital image watermarking is implemented by inserting copyright information in to the data. Different kinds of watermarking approaches are implemented namely Reversible data hiding based on histogram modification of pixel differences, Binary tree approach and Reversible data hiding considering HVS. Peak Signal to Noise Ratio (PSNR) values are calculated for different images for above mentioned approaches and corresponding PSNR Vs bits/pixel (bpp) graphs are plotted.

Since the HVS algorithm produces improved watermarked images, a public user who does not have knowledge on the original image could not identify the existence of the watermark. Thus the above algorithm is suitable to the conventional applications of the reversible data hiding, such as art, medical, and military imaging. This algorithm produces the embedded images exhibiting sharper image details compared to the original images. Therefore, even though the image enhancement is not a concern in reversible data hiding, the embedded image can replace the original image in some applications, where the sharp image details are preferred. Thus it can be used to perform the image enhancement and reversible data hiding at the same time.

10. References

[1] Fallahpour.M and Sedaagh.M.H,(2007)"High capacity lossless data hiding based on histogram modification," IEICE Electron.Exp., vol.4, no.7, pp.205-210.
[2] Hontsch.I and karam.L (2002)"Adaptive image coding with perceptual distortion control," IEEE Trans.Image Process., vol.11, no. 3, pp. 213-222.
[3] Lee.S.K, Suh Y.H and Ho.Y.S,"Reversible image authentication based on watermarking," in Proc.IEEE Int.Conf.Multimedia Expo.
[4] Lin.W, Done.L and Xue.P. (2005)"Visual distortion gauge based on discrimination of noticeable contrast changes,"IEEE Trans. Circuits syst. Video Technol., vol.15, pp.900-909.
[5] Ni.Z, Shi.Y, Ansari.N, and Su.W, (2006)"Reversible data hiding," IEEE Trans.CircuitsSyst.video Technol., vol.16, no.3, pp.354-362.
[6] Polesel.A, Ramponi.G, and MathewsV.J, (2000)"Image enhancement via adaptive unsharp masking," IEEE Trans. Image Process., vol.9, no 3, pp.505-510.
[7] Tai.W.L, Yeh.C.M, and et, al.(2009), "Reversible data hiding based on histogram modification of pixel differences," IEE Trans.Circuits Syst. Video Technol., vol. 19 pp. 906-910.
[8] Wu.M and Lin.B (2003),"Data hiding in image and video: part I-fundamental issues and solutions,"IEEE Trans. Image process, vol.12, no.6, pp.685-695.
[9] Wu.M, Yu.H, and Liu.B (2003), "Data hiding in image and video: part II- designs and applications," IEEE Trans. Image Process, vol.12, no.6, pp.696-05.