

IMPROVING VISUAL QUALITY IN REVERSIBLE IMAGE DATA HIDING USING CONTRAST ENHANCEMENT TECHNIQUES

*B. SAI SINDHU, *Research Associate*,
NEWZEN INFOTECH, Hyderabad.

ABSTRACT: A novel approach to processing digital photographs is demonstrated in this research. Reversible data concealing is the name of this technique. Maintaining a high PSNR value is not the primary objective of the proposed approach. But by emphasizing the contrasts between its components, it enhances the visual appeal of a central picture. Data is included by selecting the two biggest bins in the histogram. This allows for a second round of processing to accomplish histogram equalization. In addition to the message bits, the host image also stores other data. Because of this, we can get the original photo back to its original state. Two sets of shots demonstrate the effectiveness of the proposed strategy. So far as is known, this is the pioneering effort to enhance visual contrast through the use of reversible data hiding (RDH) techniques. Additionally, research has demonstrated that the visual integrity of images remains unaltered even after extensive addition of message bits to enhance their visibility. This outperforms three other well-known MATLAB techniques for enhancing picture contrast.

Keywords: RDH, MATLAB, Image, Histogram.

1. INTRODUCTION

Theoretical signal processing researchers have delved deeply into RDH. RDH creates a marked signal by concealing data inside a host signal; it is also known as lossless data hiding or invertible data hiding. Accurate retrieval of the original signal is possible after data extraction. If maintaining the integrity of the host signal is of the utmost importance, RDH can be a useful tool. Methods for encoding invisible data or adding watermarks to digital photos predominate in the literature.

A measure of the RDH method's efficacy is the concealment rate, which is dependent on the picture quality. A trade-off occurs when the visible content changes as the rate of concealing increases. It is less probable that embedding distortion will occur when the peak PSNR of the image histogram shifts. Current approaches regulate distributed prediction mistakes centrally and prevent data hiding distortion by using correlations between neighboring pixels. A labelled image produced using a prediction error-based technique has a decent PSNR, although the visual quality is poor due to embedding distortion. In low-light conditions, improved image quality, rather than a high PSNR, is paramount. Contrast enhancement is essential for a detailed examination of aerial and medical images. The improved picture maintains a higher level of clarity regardless of the PSNR. As far as we are aware, there is no RDH method that improves the host picture's contrast. Developing a reversible data hiding strategy that enhances contrast is the primary objective of this study, rather than merely achieving a high PSNR.

Improving an image's contrast is possible via histogram normalization. To improve contrast and add additional information, the approach modifies the histogram of pixel values.

Determine the two points in the graph that stand out the most. Each peak is divided into two adjacent bins by lengthening the bins on the borders. Separated from the peaks are the bins. Repeatedly breaking down the two largest categories in the new histogram to enhance embedding yields the intended effect of boosting contrast. By calculating the values of the edge pixels and creating a location map to accurately record their positions, you can prevent histograms from being either excessively full or too empty. The host image contains all the necessary data for restoring the original picture, including the message bits and position map. Because of this, we can extract raw data and restore all of the images. The method's efficacy was demonstrated using two sets of images. We believe this technique to be the first of its kind to use RDH to improve picture contrast. The results of the test demonstrate that compared to utilizing three different MATLAB methods for image contrast enhancement, adding a large number of message bits to contrast-enhanced images maintains the quality of the photos better.

2. OBJECTIVE

- Reversible image data concealing with contrast enhancement aims to improve the appearance of the marked image, ensure accurate data extraction, and make sure the operation can be undone completely.
- In order to guarantee that individuals are able to use the Internet.
- Through the use of an unbreakable third-party encryption mechanism, RDH ensures the secure transmission of data.

3. SYSTEM DEVELOPMENT

Using methods for reversible and lossless data embedding, the payload is digitally inserted into an image. Because it can lower image quality, data embedding is something to stay away from. Improving the appearance and clarity of photographs is achieved through reversible data embedding after data extraction. Data can be securely embedded in digital photographs via reversible data embedding, allowing access to it only by authorized users. Adaptable data-embedding methods can be evaluated based on the following criteria:

- **Data embedding capacity limit**
- **Visual quality**
- **Complexity**

The fact that reversible data embedding does not alter the data structure is one reason why it is popular. Based on the data it contains, the material will be altered. Minor shifts in pixel values can have a major impact on sensitive medical and military records. All of the data is crucial in this situation. Reversible data embedding can be utilized as a covert means of communication due to how indistinguishable the embedded and original images are. In general, there are two ways to conceal information:

Method of reversible data hiding

Method for concealing irreversible data

An irreversible method of data security

This technique for concealing modifiable data keeps the original cover and message signal intact. By using this strategy, the secret can be kept until the signal for conversation is restored. As a result, reversible data encoding has emerged.

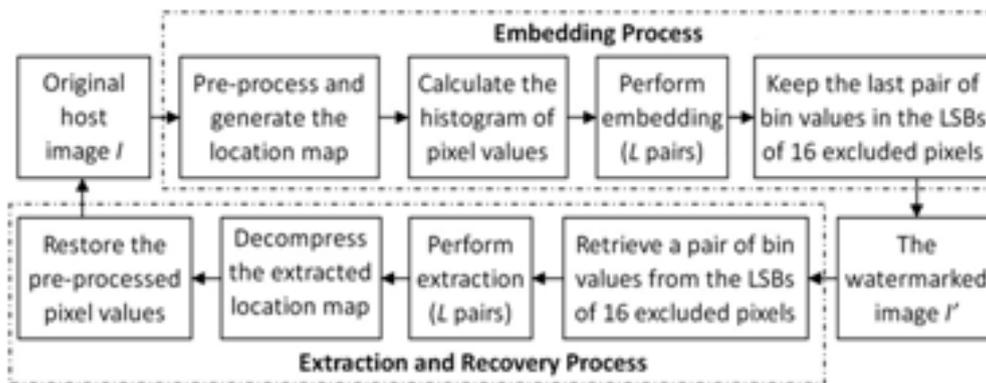


Fig. 1. The suggested RDH algorithm Matlab's pixel value matrix process for any image is:

I <256x256 uint8>										
	1	2	3	4	5	6	7	8	9	10
1	171	160	147	138	121	109	93	81	70	
2	82	73	67	64	65	72	79	90	103	
3	104	112	126	143	163	179	193	201	204	
4	205	210	209	203	193	184	170	153	133	
5	135	123	108	93	80	77	74	73	73	
6	72	81	92	103	116	131	149	162	173	
7	150	187	198	208	217	224	227	223	200	
8	165	215	209	199	182	160	149	135	111	
9	91	118	94	86	77	70	71	77	85	
10	83	85	104	112	129	142	156	166	177	
11	115	146	193	194	194	191	177	175	164	
12	127	130	150	129	118	106	91	86	81	
13	122	89	78	72	70	71	73	72	75	
14	131	103	77	74	67	63	59	53	44	
15	81	59	41	34	27	21	18	15	14	
16	21	19	18	17	17	18	17	17	17	

Total no. of pixels in image = no. of rows * no. of columns = 256 * 256
= 65536 pixels

Intensity value is calculated from the following gray scale. Where, 0 = Black, 255 = White

Fig2. Imagine yourself as a matrix

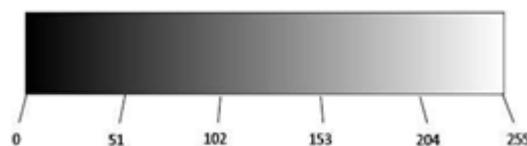


Fig 3. The grayscale scale

4. DATA EMBEDDING BY HISTOGRAM MODIFICATION

The procedure is shown with color illustrations that indicate easy stages to follow. This picture is in 8-bit grayscale. A histogram is constructed by tallying the number of pixels with grayscale values j , where j ranges from 0 to 255. $hI(j)$ is the count of j -valued pixels utilized to construct the histogram of the picture.

A. J. Here, we'll pretend that I have N numbers representing the intensities of the pixels. Then, we take the two highest values (the two highest bins) from the set of N nonempty bins $inhI$; IS denotes the smaller bin and IR the bigger. One pixel count i triggers the insertion of data. Histograms display the distribution of visual intensity frequencies. The number of pixels associated with each intensity level is displayed.

$$i' = \begin{cases} i - 1, & \text{for } i < I_S \\ I_S - b_k, & \text{for } i = I_S \\ i, & \text{for } I_S < i < I_R \\ I_R + b_k, & \text{for } i = I_R \\ i + 1, & \text{for } i > I_R, \end{cases} \quad (1)$$

Histograms display the frequency of image intensity. It displays the number of pixels for each intensity.



Fig 4. Original image

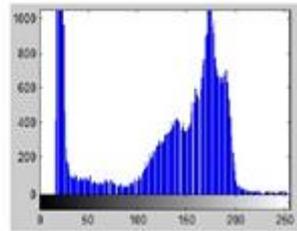


Fig 5. Histogram

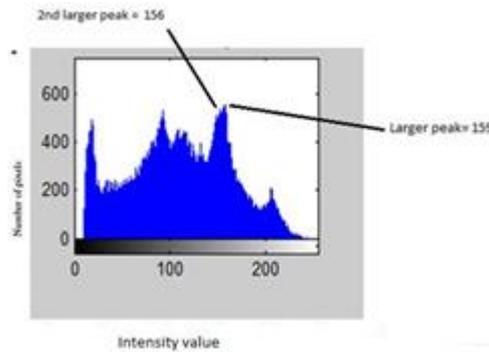


Fig 6. Histogram

A more pronounced peak is shown via infrared spectroscopy. The second-highest peak is noticeable because to its 156 Is and 159 triggering IR values.

where i represents the value of the modified pixel and b_k is the k th hidden message bit (0 or 1). Two binary values, $hI(I_S)$ and $hI(I_R)$, are assigned to each hI pixel according to Equation (1). In the modified histogram, there will be $N+2$ segments since I am missing a boundary number (0 or 255). If not, then getting ready is necessary. As they travel outward, the outer bins divide each peak into two adjacent bins, labeled as I_S-1 and I_S , I_R and I_R+1 , respectively. Both peaks' intermediate bins remain unchanged. This is a walkthrough of the Matlab process for scanning images:

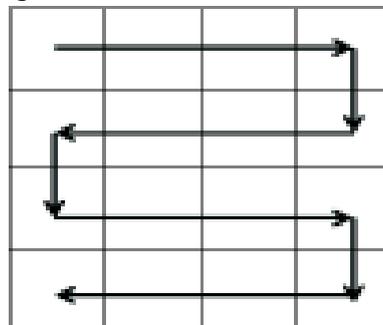


Fig 7. Scan in reverse order

Among the necessary pieces of data to be embedded are IR and Peak Is. Amount of Is and IR pairs required to retrieve embedded data. Enhance the capacity for integration. Find the term "image security." The embedded data can only be obtained by providing the peak numbers. Remove those 16 pixels from the histogram computation so you can keep them. Binary data that is concealed in images is comprised of the least significant bits (LSBs). After carrying

out the calculations in Eq. A bitwise operation substitutes 8-bit IS and IR for the 16-bit least significant bits in (1) for every data embedding pixel. After removing the 16 pixels utilized to extract the embedded data, locate the image's maximum values and generate the histogram.

$$b'_k = \begin{cases} 1, & \text{if } i' = I_S - 1 \\ 0, & \text{if } i' = I_S \\ 0, & \text{if } i' = I_R \end{cases} \quad (2)$$

$$i = \begin{cases} i' + 1, & \text{for } i' < I_S - 1 \\ I_S, & \text{for } i' = I_S - 1 \text{ or } i' = I_S \\ I_R, & \text{for } i' = I_R \text{ or } i' = I_R + 1 \\ i' - 1, & \text{for } i' > I_R + 1 \end{cases} \quad (3)$$

Afterwards, the IS-1, IS, IR, and IR+1 numbers are used to process each pixel in the histogram.

In picture I, the kth binary integer is represented by bK. It is standard practice to perform extraction and anchoring in sequential order. So says the math. (1) To restore the initial value of each histogram point, follow these steps:

To locate the initial sixteen least significant bits of the absent pixels, the restored binary values are utilized. To restore the original image, restore the deleted pixels.

Pre-Process for Complete Recovery

The procedure specifies a range of 1–254 for each hI pixel. The histogram's over/underflow behavior is determined by the value of the adjacent pixels, which can be either 0 or 255. Preprocessing the histogram before updating it will prevent this from happening. To illustrate, the numbers 0 and 255 are substituted with 1 and 254, respectively. Overflow and underflow are not conceivable since there is a finite range of values for each pixel. A location map is created with the same dimensions as the original picture. Changed pixels are given a value of 1 and unaffected pixels are given a value of 0. This process is repeated until the pre-processed pixels are recovered. This encompasses the sixteen pixels that are absent. You can make use of pre-calculated location maps stored in encrypted binary values. In order to determine which pixels were altered during extraction and recovery, data extracted from the labeled image might be used. It is possible to restore the original image by adjusting the values of these individual pixels.

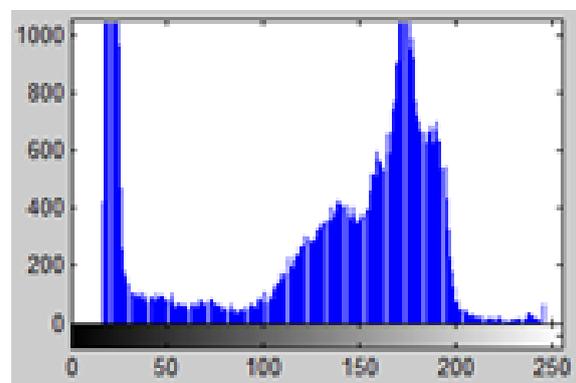


Fig 8. Histogram that has been rearranged

Changing the histogram's position prevents overflow and underflow, which is its greatest advantage. When applying grayscale or negative watermarks to photos, be careful not to underflow or overflow. Overflow, underflow, and distortions result from the current system's poor performance.

Contrast Enhancement

To create more valuable segments, Contrast Enhancement adjusts the brightness and contrast of an image's pixels. An equalization of the pixel value histogram can enhance contrast. Locate the two largest histogram bins; disregard the bins in the middle; and enlarge the outside bins to create adjacent bins for the two peaks. Both data embedding and contrast enhancement become much simpler in this manner. Increasing the number of divisions in the top two groups of the histogram enhances the contrast and insertability. The cover image stores all the necessary details for making a copy of the original image, including the position map and message bits.

Histogram equalization enhances or reduces contrast and adjusts picture intensities by adding or removing data from IR and Is. We have identified two additional peaks, Is 1 and IR.+1. As a result, data incorporation and contrast augmentation are simultaneous processes.

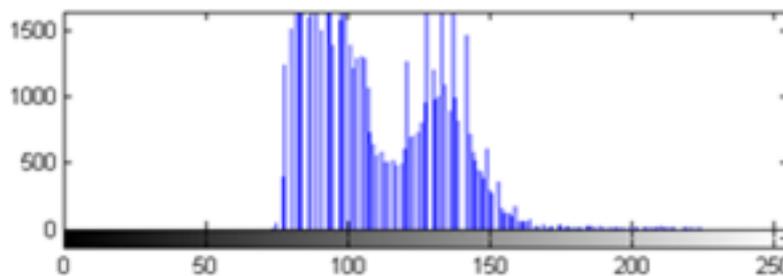


Fig 9.The first histogram

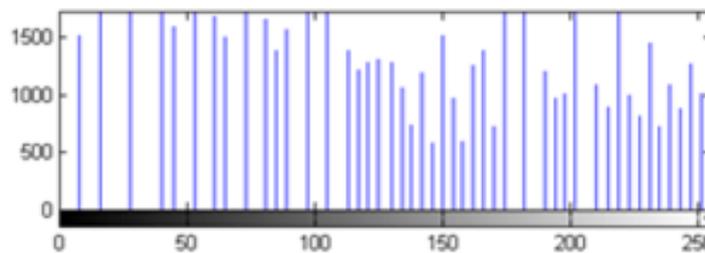


Fig.10 Histogram with Equivalents

Each of the histogram's two peaks is divided into two clusters of comparable heights because the message bits should include approximately equal numbers of 0s and 1s. The equation differentiates between the two groups formed by the two biggest bins of the adjusted histogram in order to improve the concealment rate. (1) Every single point in the cumulative distribution. In order to achieve a balanced histogram, divide each peak into two adjacent groups of equal height. As a result, data incorporation and contrast augmentation are simultaneous processes. 255 is obtained in the pre-processing stage by deducting L from 256 minus L. Once L is a positive integer, the range of pixel values is expanded to include values between 0 and L-1. A location map is created by assigning a value of 1 to modified pixels and a value of 0 to all other pixels.

The position map can be pre-calculated and compressed before being applied to the host image. L, the compressed location map's size, the earlier peak values, the two remaining peaks that require splitting, and the 16 missing pixels' least significant bits all come together. The final peak counts and associated data are obtained using Equation (2) at the separation point. The purpose of the equation is to repair and reconstruct the histogram. the third Data from previously identified peaks can be obtained via pairwise processing. Using the

recovered data, the position map is created, revealing the pixel value changes brought about by the pre-processing.

5. PROCEDURE OF THE PROPOSED ALGORITHM

- Figure 1 shows the program's steps. L sets of histogram groups must be partitioned before data can be embedded. This is accomplished by following these steps:
- Everything up to the first sixteen pixels of the bottom row is treated the same, with the exception of the pixels in the range of [0 to L-1]. A position map that is JBIG2-compressed is created in order to maintain track of these pixels.
- The picture histogram estimate does not include the bottom row's initial sixteen pixels.
- On the subject of data entry, Eq. The histogram's two highest categories, or peaks, are defined by the pixels that divide them.
- The revised histogram then separates its two peaks. Once all L pairings have been separated, the process is repeated.
- Displaying the message bits follows the bit stream of the compressed position map. The length of the compressed location map (L), the values of the peaks that came before them, the least significant bits (LSBs) of the 16 missing pixels, and the last two peaks to be separated are these components.
- The segment's peak values are substituted for the sixteen missing pixels' least important bits to create the necessary picture. The next steps, after obtaining the data and restoring the original image, are as follows:
- Looking at the least significant parts of the sixteen missing pixels allows us to locate the remaining two split peaks.
- After ensuring that the length of the compressed location map (L) is equal to the 16-bit least significant bits, data is extracted from the last two split peaks using Equation (2).
- We display missing pixels and are cognizant of high values that have already been fragmented.
- The remaining photos are processed using Equation, with the exception of the 16 reserved for recovery. Iteratively extracting and recovering the embedded data occurs until all the split peaks have been located.
- After extracting binary numbers, they are compressed and used to create the compressed location map. The decompressed image displays the preprocessed and altered pixels.
- For each pixel value below 128 we subtract L, and for every pixel value over 128 we add L. For clarity's sake and to guarantee that this condition is satisfied, the maximum L value is 64.
- The picture can be restored by rewriting the less significant parts of the sixteen pixels that were erased.

REFERENCES

1. Kede Ma, Weiming Zhang, Xianfeng Zhao, Nenghai Yu, Fenghua Li, "Reversible Data Hiding in Images by Reserving Room Before Encryption", IEEE Trans on Information Forensics and security, Vol.8, No.3, March 2013
2. Wen Chung Kuo, Po Yu Lai, Lih Chyau Wu, "Adaptive Reversible Data Hiding Based on Histogram", 10th International Conference on Intelligent Systems Design and Application, I' IEEE 2010 (2002) The IEEE website. [Online]. Available : <http://www.ieee.org/>
3. Kuo-Ming Hung, Wen-Kai Su, Ting-Wen Chen, Li-Ming Chen, "Reversible Data Hiding

- Baseon VQ and Half-toning Technique", M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High International Conference on Microelectronics, Communication and Renewable Energy (ICMiCR-2013).
4. Jose, R.; Abraham, G, "A separable reversible data hiding in encrypted image with improved performance", Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy (AICERA/ICMiCR), 2013 Annual International Conference' IEEE2013.
 5. Siddharth Malik, Anjali Sardana, Jaya, "A Keyless Approach to Image Encryption", 2012 international conference on Communication systems and Network Technologies' 2012 IEEE
 6. Yi-Hui Chen, Ci-Wei lan and ChiaoChih Huang, " A verifiable Visual Cryptography Scheme", Fifth International Conference and Evolutionary Computing' IEEE2011