

A Spatial Domain Reversible Visible Watermarking Technique for Textured Images

A. Borah
Department of CSE
Tezpur University
Tezpur, Assam, India

B. Borah
Department of CSE
Tezpur University
Tezpur, Assam, India

ABSTRACT

Digital watermarking is widely used as one of the techniques to identify ownership and protect copyrights of digital images. The watermark inserted may be visible or invisible. A Visible watermark, which is nothing but a digital pattern logo or trademark, directly convey ownership information on the media, while invisible watermarks do it indirectly after being extracted with the help of extraction techniques. A visual watermark should be such that it is clearly visible in order to survive legal procedure, yet important details of host image should not be lost. Better visibility requires higher amount of modification to the host image leading to introduction of larger amount of error and reduced *PSNR*. Visible watermarking methods are devised to embed visible watermarks by automatically determining the required optimal strength of the watermark to be inserted. One important problem with visual watermarking is that attackers can remove the watermarks by utilizing specialized algorithms like inpainting and insert their own watermarks. In order to prevent removal attack, visible watermarks having highly textured patterns or widely varying colors should be inserted into highly textured image areas. Many of the visible watermarking algorithms fail to embed watermarks in images with higher texture content. This paper proposes a reversible visible watermarking method that can deal with high texture content for the image area as well as the watermark. The watermark image with its range reduced by a factor of α is added to the host image, whose range is also reduced by a factor of $1-\alpha$. The α factor is automatically determined based on the entropy of the host as well as the watermark images. The copyright owner can easily recover the original host image with minimal error, provided information regarding the scaling factors, position of watermark and the watermark itself are preserved. Experimental results established the effectiveness of the method proposed.

General Terms

Image Processing.

Keywords

Watermark, visible, entropy, range, recovered, *PSNR*.

1. INTRODUCTION

Copyright protection of digital images becomes an important issue as images can be easily copied and distributed through the internet. One of the possible solutions for this problem is digital watermarking, which attracts lots of researchers [1]. A digital watermark is a digital pattern, logo or trademark embedded into a digital image for the purpose of protecting intellectual property right. The watermark in the watermarked image may be visible or invisible. Whichever category it is, the watermark must embed well with the host image without much loss of original image details and survives any kind of image manipulations. Digital watermarking had been proposed as a way to identify the source, creator, owner,

distributor, or authorized consumer of a document or image [2]. Watermarking has also been proposed for tracing images that have been illicitly redistributed [2]. As each buyer is uniquely marked, an illicit copy of the image itself identifies the buyer as the source.

Visible watermarks convey ownership information directly on the media and can deter attempts of copyright violations. Invisible watermarking is imperceptible to human eyes but could be extracted through extraction algorithm for identification or verification of ownership rights [3]. Embedding of watermarks, either visible or invisible, degrade the quality of the host media to some extent. Some of the required characteristics that an effective visible watermark must satisfy are as follows [4]:

- 1) A visible watermark should be obvious in both colour and monochrome images.
- 2) The watermark should spread in a large and important area of the image in order to prevent its deletion by clipping.
- 3) The watermark should be visible yet must not significantly obscure the image details beneath it.
- 4) Should not significantly obscure the image details beneath it.
- 5) The watermark must be hard to remove; removing a watermark should be more costly and labour intensive than purchasing the image from the owner.
- 6) The watermark embedding should consume little time and little effort, does not need much manual intervention.

The insertion of a visible watermark should satisfy two conflicting conditions: the watermark should be strong enough to be perceptible, yet it should be light enough to be unobtrusive and not mar the beauty of the original image [5]. Researchers face significant challenges, because of the conflicting requirements. The authors in [6] had shown that attackers can easily select watermarked areas and recover the marked regions automatically. The requirement that the watermark patterns should be clearly recognizable ensures that attackers could select the contour of the watermark easily. Furthermore, the requirement that the image details must be recognizable provides the edge information needed to appropriately propagate surrounding unmarked information. To improve the robustness of visible watermarking schemes, they suggested the following:

- (i) Embedding visible watermarks into areas containing highly textured patterns or of gradient colour distributions will undoubtedly increase the difficulty of watermark removal.

(ii) Adequately increasing the complications of shapes or textures of embedding patterns will increase the difficulty to correctly select the watermarked area.

(iii) Fragile invisible watermarks can be used to guard the integrity of visibly embedded images.

Spatial as well as frequency domain techniques are available for watermarking. Basic principle used in visible watermarking is to add the watermark image at some specific positions of the host image after scaling the images using suitable scaling factors. Available methods [1-17] vary in selecting the scaling factors and adaptive positioning. Reversibility of the watermarking is also a major concern [9-14]. Some methods like [13] do not use the concept of scaling factors.

Many of the visible watermarking algorithms fail to embed watermarks in images with higher texture content. This paper proposes a reversible visible watermarking method that can survive attempts for removing it. The method can embed highly textured watermark images into highly textured regions of host images by automatically adjusting the needed strengths and locations to be inserted. The method works by manipulating the ranges of the watermark as well as the host images based on entropy as texture measure. Experimental results reveal that the method works without much degradation of the original content.

2. BACKGROUND CONCEPTS

Some fundamental concepts regarding images are presented below.

2.1 Digital Image Fundamentals

An image having M rows and N columns is a two-dimensional function $f(x, y)$, $x=0,1,2,\dots,M-1$, $y=0,1,2,\dots,N-1$, where, amplitude of the function f at any pair of coordinates (x, y) represents intensity (brightness) or gray level of the image at that point. For digital images x, y and f are finite discrete quantities. Each element of the image identified by a particular location and intensity value is called a pixel. Each pixel may have discrete intensity levels in the range $[0, L-1]$, such that $L=2^k$, where k is an integer. For visual images $k=8$, requiring one byte to store each pixel value, which is in the range $[0, 255]$. In color images each pixel is described by a vector of three numbers corresponding to the intensity of the three primary colors red(R), green(G), and blue(B). Besides the R, G and B channels, a color pixel may have a fourth value known as the alpha channel, which indicates the degree of transparency (or opacity) of the pixel. It specifies how the pixel's colors should be merged with another pixel when the two are overlaid, one on top of the other. The more opaque a pixel, the more it hides the background against which the image is presented. Zero alpha represents a completely transparent pixel, maximum alpha (255) represents a completely opaque pixel, with levels in between representing the degree of haziness. Thus, a color pixel requires 4 bytes to store.

Images are stored in image file formats as a one dimensional array called the raster data. A typical image file format such as TIFF, PNG, JPEG, BMP, GIF etc. has two parts, the header containing the magic number and the raster data. The magic number identifies the file format, the image dimensions, and the compression technique used to store the raster data. Some file formats (e.g. BMP) do not use any compression technique requiring larger space to store the image. Others use lossless (e.g. TIFF, PNG) lossy (e.g. JPEG) compression techniques. In some formats (e.g. GIF) the image is saved using an index

number corresponding to the color palette that is stored in the image header.

The PNG format is widely used for transferring images on the Internet. It supports palette-based images with palettes of 24-bit RGB or 32-bit RGBA colors, grayscale images with or without alpha channel, and full-color non-palette-based RGB[A] images with or without alpha channel.

2.2 Image Texture measurement

Images are characterized by properties such as color, texture etc. Variation in intensity and color of the pixels results in the appearance of texture. Although no formal definition of texture exists, intuitively this descriptor provides measures of properties such as smoothness, coarseness, and regularity [18]. Texture analysis can be done in three principal approaches: statistical, structural and spectral. Statistical approaches characterize texture into categories like smooth, coarse and grainy. Structural techniques utilize spatial arrangement of image primitives such as regularly spaced parallel lines to model texture. Spectral techniques use properties of the Fourier spectrum and used primarily to detect global periodicity in an image by identifying high-energy, narrow peaks in the spectrum.

One simple measure used for texture computation in statistical approach is moments of the intensity histogram of an image or region. Another measure based on histogram is the entropy measure. Gray level co occurrence matrix method is a way of extracting second order statistical texture measures.

2.2.1 The Entropy Measure

Let z be a random variable denoting intensity and let $p(z_i)$, $i=0, 1, 2, \dots, L-1$, be the corresponding normalized histogram, where L is the number of distinct intensity levels. Entropy $e(z)$ is defined as:

$$e(z) = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$$

The normalized histogram element, $p(z_i)$ represent the probability of occurrence of i -th gray level z_i of the image. The p_i s have values in the range $[0, 1]$ and their sum equals 1. If the image has M rows N columns and gray level z_i occurs n_i number of times, then

$$p_i = \frac{n_i}{MN} \quad i = 0, 1, 2, \dots, L-1$$

When probability distribution is uniform, entropy becomes the maximum [19]. In this case all intensity levels occur equal number of times having the same probability, given by

$$p_i = \frac{1}{L} = 2^{-k}$$

The maximum entropy is found from

$$e(z)_{\max} = -\sum_{i=0}^{L-1} \frac{1}{L} \log_2 \frac{1}{L} = -\log_2 \frac{1}{L} = k$$

For visual images with 256 gray levels maximum entropy is 8. When all pixels of the image have the same gray value, $p_i = 1$ and entropy becomes zero, the minimum.

2.3 Color to Grayscale Conversion

Grayscale images are still in demand for various image processing tasks. A coloured image can be converted to a grayscale image using various methods [20]. For each colour pixel described by a triple (R, G, B) a single number,

$Z=aR+bG+cB$ is to be derived giving a grayscale value for the pixel. Here the constants a, b, c are such that $a+b+c=1$. The average intensity method simply averages the three values giving $Z=1/3R+1/3G+1/3B$. Instead of equal weights, variable weights can be used as human eyes are more sensitive to green than other colours. The luminosity method computes gray scale equivalent of a coloured pixel using weighted averaging as:

$$Z = 0.299R + 0.587G + 0.114B$$

3. PROPOSED ALGORITHM

Let $f_1(x, y), x=0,1,2,\dots,M_1-1, y=0,1,2,\dots,N_1-1$ represent the host image, where the watermark image $f_2(x, y), x=0,1,2,\dots,M_2-1, y=0,1,2,\dots,N_2-1$ need to be inserted. Let, $f_3(x, y), x=0,1,2,\dots,M_1-1, y=0,1,2,\dots,N_1-1$ represent the resultant watermarked image, which is of the same size as the host image. Sizes for the three images are $M_1 \times N_1, M_2 \times N_2$ and $M_1 \times N_1$. Subsequently, an image $f_i(x, y)$ will be represented simply by f_i and it will be used to mean either a particular pixel of the image or the image itself. Note that subscripts 1, 2 and 3 denote the host, watermark and watermarked images respectively.

3.1 Scaling factors for images

The range of the watermark image needs to be made narrower so that it does not obstruct the details of the host image significantly. Let the watermark image pixel intensity values occur in the range $[min_2, max_2]$. This image is to be converted to a new image with a narrower range $[newmin_2, newmax_2]$. The pixel value f_2 becomes f_2' in the new range. Then,

$$\frac{f_2 - \min_2}{\max_2 - \min_2} = \frac{f_2' - \text{newmin}_2}{\text{newmax}_2 - \text{newmin}_2}$$

Take the original range to be $[0, 255]$ and the new narrower range to be $[0, S]$ with $S < 128$. Putting $min_2=0, max_2=255, newmin_2=0, newmax_2=S$ in the above equation results in:

$$\frac{f_2 - 0}{255 - 0} = \frac{f_2' - 0}{S - 0}$$

$$\Rightarrow f_2' = \frac{S}{255} f_2 = \alpha f_2$$

Where, the constant α ($0 < \alpha < 0.5$) is the scaling factor of the watermark image.

This transformed image is to be added to the host image. It is better to reduce the range of the portion of host image covered by the watermark, so that no pixel value of the watermarked image exceeds 255 after adding the watermark image to it. Let, the original range for the host image be $[0, 255]$ and the new reduced range for the watermarked area be $[0, 255-S]$. A host image pixel value f_1 in the original range becomes f_1' in the new range. The calculation of f_1' is similar to the calculation of f_2' :

$$\frac{f_1 - 0}{255 - 0} = \frac{f_1' - 0}{255 - S - 0}$$

$$\Rightarrow f_1' = \frac{255 - S}{255} f_1 = f_1 \left(1 - \frac{S}{255}\right)$$

$$\Rightarrow f_1' = f_1(1 - \alpha) = f_1\beta$$

Where, $\beta=1-\alpha$ is the scaling factor of the host image. Thus, the watermarked image is obtained as:

$$f_3 = \beta f_1 + \alpha f_2$$

3.2 Determining values of Scaling factors

The value of scaling factors are derived based on the entropy (e_2) of the watermark as well as entropy (e_1) of the portion of host image where the watermark is inserted. The average intensity (μ_2) of the watermark image and the average intensity (μ_1) of the corresponding portion of host image also take part in determining the values of scaling factors. Recall that scaling factor of watermark image is $\alpha = S/255$ and scaling factor of host image is $\beta=1-\alpha$. Therefore, S should be determined first, using $S=S_1+S_2$, where, S_1 is the contribution of the host image and S_2 is the contribution of watermark image. In order to calculate the entropy and average intensity of pixels, a colored image is converted to a grayscale image. The luminosity method can be used to perform this conversion.

Determination of the scaling factors involves the following steps:

1. Find grayscale equivalents for color image f_2 and calculate average intensity μ_2 using

$$Z_2(x, y) = 0.299R_2(x, y) + 0.587G_2(x, y) + 0.114B_2(x, y)$$

$$x = 0,1,\dots,M_2 \text{ and } y = 0,1,2,\dots,N_2$$

$$\mu_2 = \frac{1}{M_2 N_2} \sum_{x=0}^{M_2} \sum_{y=0}^{N_2} Z_2(x, y)$$

2. Find grayscale equivalent for the portion of f_1 covered by f_2 and the average intensity, μ_1 using calculations similar to step 1 above. If (x_0, y_0) is the location in f_1 for watermark insertion, then $Z_1(x, y) = 0.299R_1(x, y) + 0.587G_1(x, y) + 0.114B_1(x, y)$
 $x = 0,1,2,\dots,M_1$ and $y = 0,1,2,\dots,N_1$

$$\mu_1 = \frac{1}{M_2 N_2} \sum_{x=x_0}^{x_0+M_2} \sum_{y=y_0}^{y_0+N_2} Z_1(x, y)$$

3. Calculate entropy e_2 of watermark p_2 (section 2.2.1)
4. Calculate entropy e_1 of portion of p_1 covered by watermark (section 2.2.1)
5. Calculate S_2 using:

$$S_2 = 1 + \log_2(1 + e_2)^2$$

6. Adjust S_2 based on μ_2 :

$$S_2 = S_2 + S_2 * \frac{255 - \mu_2}{128} * \frac{2^{e_2}}{128}$$

7. Calculate S_1 based on e_1 :

$$S_1 = 1 + \log_2(1 + e_1)^2$$

8. Adjust of S_1 based on μ_1 :

$$S_1 = S_1 + S_1 * \frac{255 - \mu_1}{128} * \frac{2^{e_1}}{128}$$

9. Calculate $S = S_1 + S_2$.

10. Calculation of α, β :

$$\alpha = \frac{S}{255}$$

$$\beta = 1 - \alpha$$

In steps 5 and 7 addition of the constant 1 ensures minimum value of 1 for S_2 and S_1 . Presence of 1 in $\log_2(e_2+1)$ and $\log_2(e_1+1)$ is required to avoid calculating $\log_2(0)$ when $e_2=0$ or $e_1=0$.

3.3 Inserting the Watermark

Insertion of watermark becomes simple once α and β are determined. It is done using the following equation:

$$f_3 = \lfloor \beta f_1 + \alpha f_2 + 0.5 \rfloor$$

For color images, the same scaling factors are applied for each of the three color channels giving:

$$R_3 = \lfloor \beta R_1 + \alpha R_2 + 0.5 \rfloor$$

$$G_3 = \lfloor \beta G_1 + \alpha G_2 + 0.5 \rfloor$$

$$B_3 = \lfloor \beta B_1 + \alpha B_2 + 0.5 \rfloor$$

Then, R_3 , G_3 , B_3 are combined to form a single color pixel. In order to make removal difficult, the watermark should be inserted in a region having high entropy. So, entropies of different regions of the host image are calculated and the region with highest entropy is selected. The sizes of the regions considered should be equal to the size of the watermark. Each position can be identified by the coordinates of the upper left corner. In order to prevent attackers from easily identifying position of the inserted watermark, search for the highest entropy region should start at a random location (x_0, y_0) at the top left corner of the host image. Then new regions are selected by adding a fixed horizontal and vertical shift to the previous position. After examining the whole image in this manner the position with highest entropy is selected for inserting the watermark. This location may not be the globally best location possible.

3.4 Recovering the Host Image

Given the watermarked image, the copyright owner can remove the watermark and restore the original image, provided the information regarding the scaling factors, position of the watermark and the watermark itself are available. A minimal error will be introduced due to rounding during the watermarking process. It can be proved that the absolute error introduced per pixel will be at most 1. Reversing is done as follows:

$$r_R_1 = \left\lfloor \frac{R_3 - \alpha R_2}{\beta} + 0.5 \right\rfloor$$

$$r_G_1 = \left\lfloor \frac{G_3 - \alpha G_2}{\beta} + 0.5 \right\rfloor$$

$$r_B_1 = \left\lfloor \frac{B_3 - \alpha B_2}{\beta} + 0.5 \right\rfloor$$

here, r_R_1 , r_G_1 and r_B_1 are the R , G and B component of the recovered host image pixels.

4. EXPERIMENTAL RESULTS

Experiments are performed on several images downloaded from the USC-SIPI image database [21] and wallpaperswide.com. Programming is done in Java SE platform. Images in the PNG file format are considered. Source images are converted to PNG format if needed.

The watermarked image (f_3) is evaluated by calculating mean square error (MSE) and peak signal-to-noise ratio ($PSNR$) with respect to the host image (f_1). These quantities are defined for grayscale images as follows:

$$MSE = \frac{1}{M_1 N_1} \sum_{x=0}^{M_1-1} \sum_{y=0}^{N_1-1} [f_3(x, y) - f_1(x, y)]^2$$

$$PSNR = 20 \log_{10} \left\{ \frac{255}{\sqrt{MSE}} \right\}$$

In case of colored images the quantities can be calculated separately for each of the color components and average can be taken. High value of $PSNR$ is desirable. The error is involved only with the watermarked regions of the host image and remaining portion is error free. Therefore MSE and $PSNR$ can be calculated with respect to watermarked regions of the host image only [13]. In this case the terms $QMSE$ and $QPSNR$ are used. The modified definitions are:

$$QMSE = \frac{1}{n M_2 N_2} \sum_{x=0}^{M_2-1} \sum_{y=0}^{N_2-1} [f_3(x, y) - f_1(x, y)]^2$$

$$QPSNR = 20 \log_{10} \left\{ \frac{255}{\sqrt{QMSE}} \right\}$$

where, n is the number of copies of watermark of size $M_2 \times N_2$ inserted in the host image.

Consider that only one copy of the watermark is inserted and uniform error involved in each pixel in the watermarked region is at most one. Then, $PSNR=48$ is obtained. An error of 1 per pixel will not be detectable by the human eye. Therefore $PSNR$ greater than 48 means that error involved is negligible. But, visible watermarks should be clearly recognizable requiring $PSNR$ to be lower than 48. Now suppose average error involved per pixel is 25.5, then computed error measure becomes, $PSNR=20$. Highly textured images require larger alterations for better visibility of the inserted watermark. For lowly textured images, $PSNR$ greater than 30 may also be acceptable. So, for visible watermarking $PSNR$ in the range 20-40 are reasonable.

Experiments are done involving three watermark images listed in Table 1 and 10 host images listed in Table 2. The watermark images are shown in Figure 1. The host images before watermarking are not displayed here in order to save space. Entropy and average intensity values are calculated after finding grayscale representation of the colored images.



Fig 1: Images used as watermarks

Table 1. Watermark image properties

Image name	Size	Entropy	Average intensity	Source
TUlogo	225×225	5.468	131.034	Institute logo
Girl	256×2256	7.053	58.300	SIPI
Rupee	337×149	7.337	139.908	Downloaded

Table 2. Host image used in the experiments

Image no.	Image name	Size	Source
1	bandon	610×403	SIPI
2	lena	512×512	SIPI
3	opera	695×586	SIPI
4	capcicum	512×512	SIPI
5	baboon	512×512	SIPI
6	newyork	842×571	SIPI
7	1.1.06	512×512	SIPI
8	yosemite	800×600	Wallpapers
9	lion_32	800×600	Wallpapers
10	elaine	512×512	SIPI

The watermarked images are displayed below in Figure 2-4. The host images appear in these figures in the same order as displayed in Table 2. Properties of different watermarked images (Figure [2-4]) with TUlogo, Girl and Rupee images embedded are shown in Table 3, Table 4 and Table 5 respectively. The inserted watermarks are removed by applying the reversing algorithm. Computation of *PSNR* is also done for the recovered images with respect to the original host images and recorded along with *PSNRs* of the watermarked images. *PSNR* values are calculated by taking the average value of *PSNRs* of Red, Green and Blue color channels. *QPSNR* values are also computed and shown in the tables.

It can be seen from the tables [2-4] that *PSNRs* of the watermarked images remain near about 30. Region based *QPSNR* values, which are lower than *PSNR* values, remains greater than 20. The reverse *PSNR* for the images are not less than 55 indicating that the error involved in recovering the host images is negligible, impossible to detect by human eye. Also, it was observed during experiments that error involved in each pixel of the recovered host images is either 0 or 1. The watermarks cannot be destroyed by image manipulations such as smoothing or zooming.

Table 3. Watermarked image properties (TUlogo inserted)

Image no.	entropy	Avg. Intensity	PSNR	Q PSNR	Rev. PSNR
1	7.19	96.39	28.95	22.09	57.18
2	7.45	125.65	32.23	25.09	56.62
3	7.22	117.40	35.53	26.47	56.94

Table 4. Watermarked image properties (Girl inserted)

Image no.	entropy	Avg. Intensity	PSNR	Q PSNR	Rev. PSNR
4	7.36	128.49	30.47	24.45	55.29
5	7.21	141.40	28.04	22.02	55.47
6	7.84	100.73	30.63	21.98	54.43

Table 5. Watermarked image properties (Rupee inserted)

Image no.	entropy	Avg. Intensity	PSNR	Q PSNR	Rev. PSNR
7	6.73	129.58	34.33	27.17	56.68
8	7.61	126.42	32.29	22.70	55.80
9	7.70	110.15	31.89	22.10	55.38

In Figure 6, results are presented when different watermarks are inserted one by one into the same image. Corresponding evaluation measures are placed in Table 6. Recovered host images obtained by applying the reversing algorithm are also displayed in Figure 6. No difference in appearance can be noticed in the three recovered images.

Table 6. Watermarked image properties (same image)

Image no.	entropy	Avg. Intensity	PSNR	Q PSNR	Rev. PSNR
10	7.08	134.11	33.47	26.33	57.18
10	7.10	130.62	28.90	22.89	55.46
10	7.07	119.00	32.65	25.49	56.33

From the experimental results it can be seen that the proposed algorithm is able to insert visible watermarks in a translucent manner without obscuring the host image details to a larger extent. Moreover the owner can remove the watermarks inserted and recover the host images without loss of any visual quality.

5. CONCLUSION

In this paper a visible watermarking technique is presented that automatically selects a highly textured region of the host image to insert the watermark in a translucent manner so that the watermark as well as the host image details in the covered area remains clearly visible. The purpose of selecting a highly textured region is to prevent attackers from removing the watermark easily. Texture measurement is done based on entropy. Entropy of a coloured image is calculated by finding gray scale equivalent of the image. Main contribution of the paper is the calculation of needed strength of the watermark suitable for the particular situation. A nonlinear measure involving entropy of both host and watermark images is used for this purpose. Experiments performed on several images show that the method really works.

The RGB colour model is used in the algorithm. Other visually uniform colour models may produce even better results. Another possible area for improvement may be using a texture measure other than entropy. Further experiments may be performed in these directions together with detailed attack analysis.

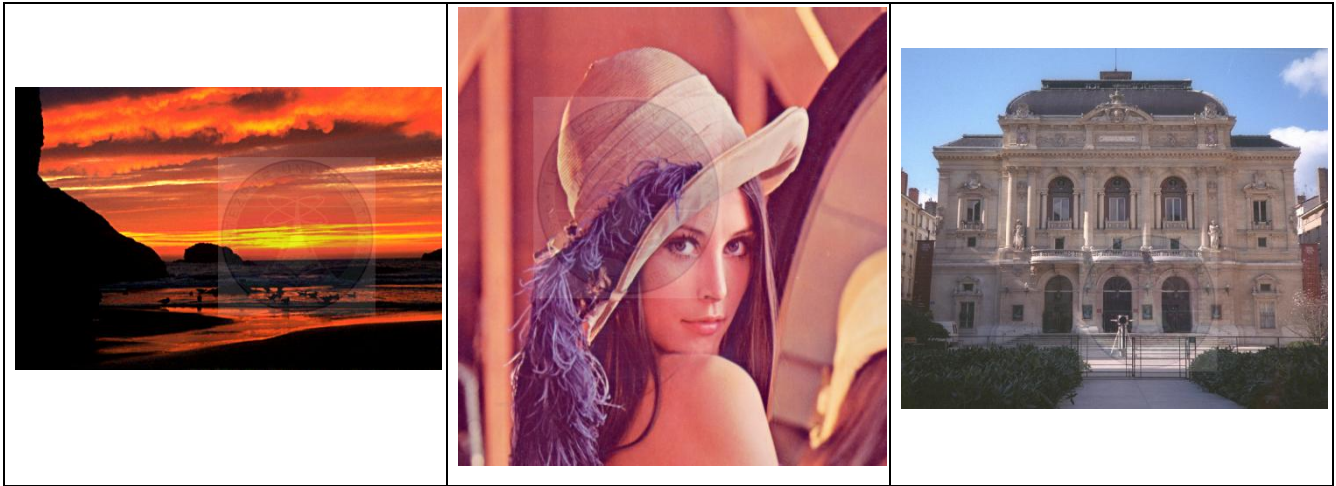


Fig 2: Images watermarked with TUlogo

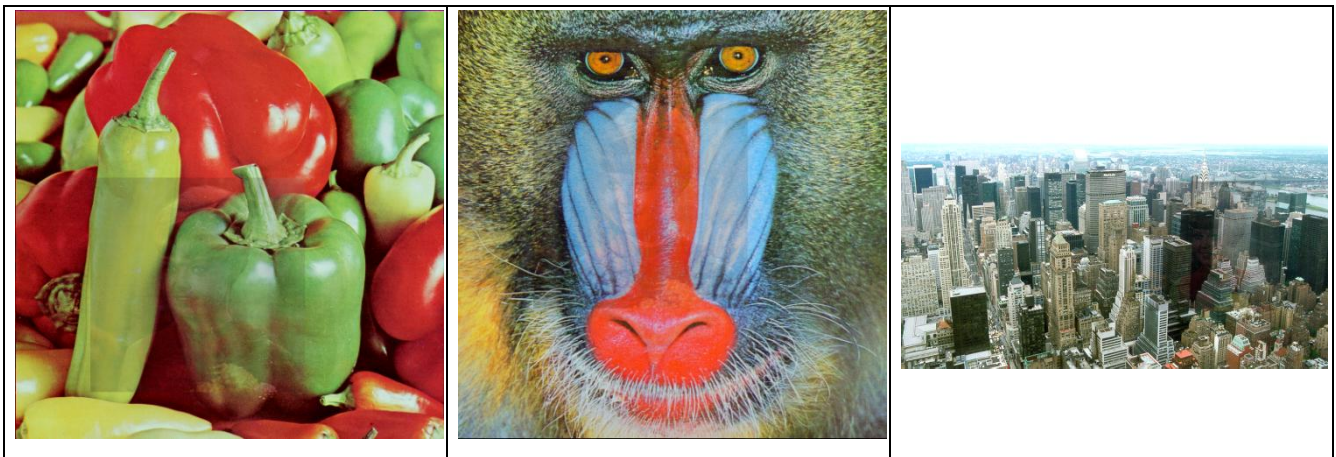


Fig 3: Images watermarked with girl

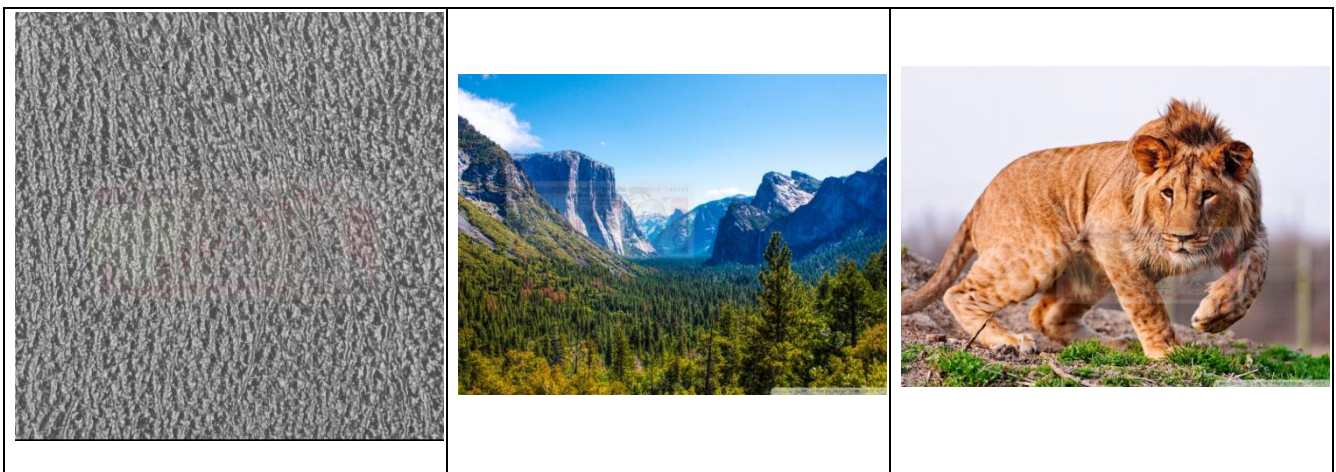


Fig 4: Images watermarked with rupee

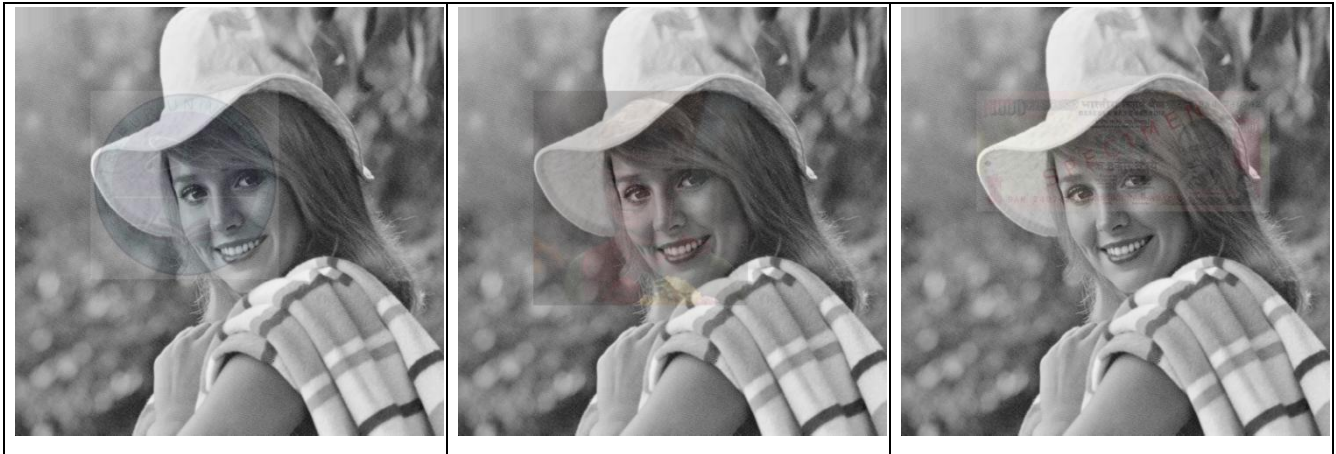


Fig 5: Elaine image watermarked with TUlogo, girl and rupee

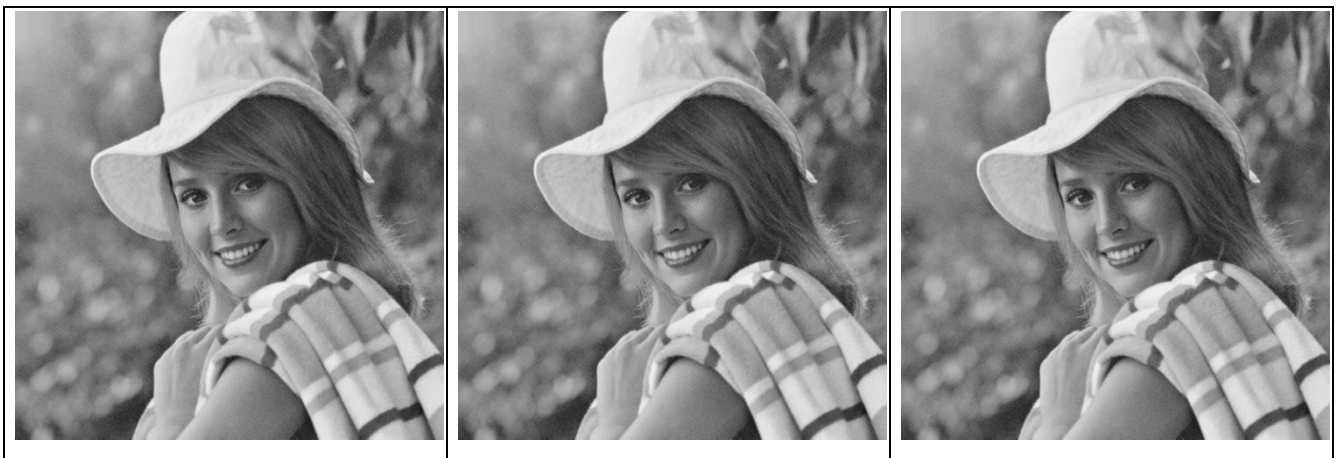


Fig 6: Images obtained after removing watermarks from watermarked Elaine image

6. REFERENCES

- [1] Chen P.-M. 2000. A visible watermarking mechanism using a statistic approach. Proc. of WCCC-ICSP 2000, pp. 910-913.
- [2] Berghel H. and O’Gorman, L. 1996. Protecting ownership rights through digital watermarking. IEEE Computer Mag, pp. 101-103.
- [3] Kankanhalli M. S., Rajmohan and Ramakrishnan K. R. 1999. Adaptive visible watermarking of images. In Proc. International Conference on Multimedia Computing and Systems, Vol. 1, IEEE, pp.568-573.
- [4] Zeng W. and Yanpeng W. 2010. A visible watermarking scheme in spatial domain using HVS model. Information Technology Journal 9(8), 2010, pp. 1622-1628.
- [5] Rao, A. R., Braudaway G. W. and Mintzer F. C. 1998. Automatic visible watermarking of images. Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques-2, vol. 3314, pp. 110–121.
- [6] Huang C.-H. and Wu, J.-L. 2004. Attacking visible watermarking schemes. IEEE Transactions on Multimedia, vol. 6, no. 1, pp. 16–30.
- [7] Lumini A. and Maio D. 2004. Adaptive positioning of a visible watermark in a digital image. IEEE International Conference on Multimedia and Expo, ICME 2004 , vol. 2, pp. 967-970.
- [8] Huang B.-B. and Tang S.-X. 2006. A contrast-sensitive visible watermarking scheme, Multimedia, IEEE 13(2), pp. 60-66.
- [9] Hu, Y., Kwong S. and Huang J. 2006. An algorithm for removable visible watermarking, IEEE Transactions on Circuits and Systems for Video Technology., vol. 16, no. 1, pp. 129–133.
- [10] Hu, Y. and Jeon B. 2006. Reversible visible watermarking and lossless recovery of original images. IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 11, pp. 1423–1429.
- [11] Tsai H.-M. and Chang L.-W. 2007. A high secure reversible visible water marking scheme. In Proc. IEEE Int. Conf. Multimedia Expo, 2007, Beijing, China, pp. 2106–2109.
- [12] Yang, Y., Sun, X., Yang, H., Li, C.-T. and Xiao, R. 2009. A contrast-sensitive reversible visible image watermarking technique. IEEE Transactions on Circuits and Systems for Video Technology, 19(5), 2009, 656–667.
- [13] Liu T.-Y. and Tsai W.-H. 2010. Generic lossless visible watermarking – A new approach. IEEE Transactions on Image Processing, 19(5), 1224-1235.
- [14] Zhang, X. , Wang, S. and Feng, G. 2011. Reversible Visible Watermarking with Lossless Data Embedding

Based on Difference Value Shift. *Intelligent Automation & Soft Computing* 17.2, pp. 233-243.

- [15] Kalra, G. S., Talwar, R. and Sadawarti, H. 2011. Protecting Copyright Multimedia Files by Means of Digital Watermarking: A Review. *The IEEE International Conference on Advanced Computing and Communication Technologies [ICACCT-2011]* ISBN 81-8788503-3. <http://www.apiit.edu.in/downloads/all%20chapters/CHAPTER-55.pdf>.
- [16] Lee, C-W and Tsai, W-H. 2012. A New Lossless Visible Watermarking Method via The use of The PNG Image.
- [17] Jiang, M. 2012. A new adaptive visible watermarking algorithm for document images. *Information Technology Journal*, 11(9), 1322-1326.
- [18] Gonzalez, R. C. and Woods, R. E. 2009. *Digital Image Processing*, Third edition, PEARSON.
- [19] <http://www.johnloomis.org/ece563/notes/basics/entropy/entropy.html>.
- [20] Qiu, M., Finlayson G. D. and Qiu, G. 2008. Contrast Maximizing and Brightness Preserving Color to Grayscale Image Conversion. *CGIV2008, 4th European Conference on Colour in Graphics, Imaging, and Vision*, June 2008, pp. 347-351, Terrassa-Barcelona, Spain.
- [21] Weber, A. G. 1997. *The USC-SIPI Image Database: Version 5*, October 1997, Signal and Image Processing Institute, University of Southern California.