Useful Information Embedding in Images Using Watermarks

Sunil Mohan Adapa
International Institute of
Information Technology, Hyderabad
mohan@gdit.iiit.net

Jayanthi Sivaswamy
International Institute of
Information Technology, Hyderabad
jsivaswamy@iiit.net

Abstract

Watermarking is being used in a wide variety of applications. Steganography, copyright protection, owner identification etc are some of them. But watermarking can also be used as means to store other kind of useful information in the image. This work discusses the advantages of putting such information into the image. A watermarking algorithm suitable for embedding large amount of information in the image, robust of jpeg compression is also presented.

1. Introduction

Images convey more information than text, but images do not say everything. Additional information is always associated with an image. This information is sometimes put into the image as a comment. This is no better than keeping the information in a separate file or a different column in a database. Let us take a geographical map for example. Various places in the map are to be labeled. If one wants to do this without disturbing the original image, the solution is to keep the information separately and not etch it over the map. But if one crops the image to a particular section then will the information about that section be retained and the information about other sections of the map be lost? The solution to this is to keep the information as close to the image as possible. The information about a section of the image should preferably be stored in that same section of the image.

In medical applications, it may be desirable to keep the diagnostic and general information about the patient close to the image. Anatomical and other information pertaining to the image can also be included following this method.

Similarly, preprocessed data in computationally intensive image processing applications can be kept in the image to be used by the application to avoid repeated preprocessing. Features vectors obtained from an image can be stored in the image and applications like image databases can directly use them. Watermarking provides a solution to the above and can help to store relevant information. Watermarking is a widely used technique to embed information in an image. However, most of its use to date is with the objective of authenticating and establishing the true ownership of an image. There are a number of distinct application areas for watermarking. A brief discussion of them is presented in [3, 6, 2]. In steaganography, watermarks are used to embed highly secure information in images. In this paper, we examine the possibility of using watermarks for storing useful information and propose some solutions. We first summarise the different exisiting applications for watermarks. In section 2 we examine the requirements on a watermark to store useful information. In section 3 we propose some methods for embedding useful information in watermarks. These methods attempt to meet the requirements outlined in section 2.

1.1. Copyright Protection and Fingerprinting

As an example of this application consider the following. Digital images of paintings from a museum and satellite images need copyright protection to prevent piracy. Ownership of digital media can be verified in the case of a copyright dispute by using the embedded data as a proof. The requirements on the watermark for this purpose are that the watermark be robust and be tolerant to malicious and unintentional attacks on the watermarked image. A number of watermarking techniques proposed for these applications require the original image to be available during the watermark detection phase [5, 1]. These watermarking methods are called private schemes. Most of these methods are aimed at just “detecting” the watermark and verifying the ownership. They are not suitable to transmit useful data along with the image. Despite their advantages and robustness, such methods are limited by the requirement that the original image be known during the detection stage.

1.2. Authentication or Tamper-proofing

Many applications need to confirm the integrity of images and detect any tampering. For instance, during criminal investigation, the law enforcing authorities would like to be sure that the digital images that they are using are not tampered with maliciously. Also personal identification records of individuals like employee record in a company and passport photographs require a tamper-proof arrangement to ensure the integrity of the media. When an image is transmitted over a channel, there is no guarantee that the received image is the unmodified original image. The task of authentication and tamper-proofing is to report any such changes in the image [4]. Unlike the previous application, the watermark embedded in an image should change with any deliberate manipulations of the image for alerting any tampering in authentication type of applications. Such watermarks are fragile watermarks.

1.3. Steganography

Steganography is another area where ”secret” messages are embedded in images and other type of data to prevent their detection. This is typically used in scenarios where local channels have to be used to communicate which may not be secure. Hence the message should be sent in such a way that the presence of the message is not revealed. Watermarking methods can be used for such hidden transmission to ensure that the presence of the message in the image is undetectable. Watermarking methods meant for steganography can be used for embedding high amount of information. But most of them are not robust to compression. These methods are also fragile as robust watermarks are easily detectable.

1.4. Captioning and Annotation

These watermarking methods aim at embedding of descriptive information in the image. These methods are usually fragile and are not robust to compression, cropping etc. However, unlike other applications, this application requires moderately large information bearing capacity for the watermark. Thus while other watermarking methods have information capacity of the order of few tens of bytes, this application can have much higher capacity. To sum up, none of the above methods are adequate for our purpose. They suffer from one or more of these drawbacks: insufficient information capacity, require of access to the original image during detection or lack of robustness to compression.

2. Requirements for Embedding Useful Information

Embedding useful information in watermarks imposes certain specific requirements on the watermark. We now examine these special needs. A watermarking algorithm that is used in embedding useful information should ideally have the following properties.

1. The watermarking method should not require the original image to be present during the watermark extraction phase. The watermarking method then falls in the category of “blind” watermarks. Blind watermarks usually lack robustness.

2. A very important property that the watermark should possess is robustness to compression specifically to jpeg since it is widely used as an image format. The watermark should adapt itself to various compression qualities used by jpeg.

3. The watermark must have a considerably large (several kilobytes) information capacity. The size of the watermarked image should not unduly increase with more and more watermark content in it. The watermark capacity and the size of an image should be fairly unrelated.

4. The watermarking information about a part of the image should be embedded only in that part of the image. When the watermarked image undergoes operations such as cropping, the information that is associated with the selected part of the image should be retained and the information that concerns other parts should be removed.

5. For most applications, the watermark should not introduce visible distortion in the image. Medical applications are the most sensitive. For some applications some distortion in the image content might be acceptable. For example in copyright protection, slight distortion might be allowed when the watermark is robust.

It should be noted that many of these properties are mutually exclusive.

3. Proposed Methods for Information Embedding

We now propose some methods to embed useful information. These methods have been designed to meet the above requirements by and large.

3.1. A Simple Method

One simple method that is not robust to compression, using jpeg or other methods, is the LSB encoding. LSB encoding is very simple and has been used for a variety of purposes. In this method the last significant bit of every component (or the blue component, that is least observable) is replaced by the watermark information bit. This method has considerable information capacity, but the amount of information that can be embedded is still limited.

Figure 1 shows LSB encoding scheme.

3.2. Methods that are robust to compression

3.2.1 Jpeg compression

Since watermarking methods that are robust to compression and have considerable information capacity are highly de-

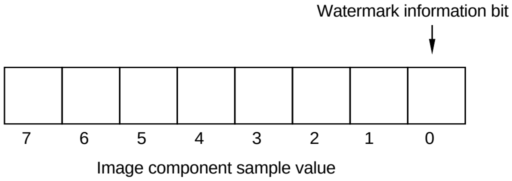


Figure 1: LSB encoding

sirable, we have considered the widely used jpeg compression scheme to study if a watermarking scheme can be incorporated in the compression scheme. We begin with a brief description of the jpeg scheme. The jpeg encoding and decoding procedures consist of several steps. In the encoding process, the source image is first divided into 8x8 blocks. The forward DCT coefficients are computed for each of these blocks and quantised prior to entropy encoding. To achieve compression the following property of the DCT is exploited: There is a one to one mapping from the original samples to the DCT coefficients. DCT has the property that most of the coefficients are significant only for the lower frequencies while the higher frequency coefficients are close to zero. Hence, these low value coefficients need not be encoded. In the jpeg scheme, the DCT coefficients are fed to the quantizer which divides each coefficient with a corresponding value from a quantization table specified by the application. This step makes sure that the coefficients are represented with no more precision that required. It discards all the information from the image that is visually unnecessary. The retained quantized DCT coefficients are finally entropy coded. Entropy encoding further compresses the coefficients based on their statistical nature. Huffman coding is usually employed in this step. For decompression, the steps are reversed. The compressed image is entropy decoded first to recover the quantized coefficients. These are dequantised next and finally IDCT is applied to reconstruct the spatial domain image. At the decoder's end, the same quantization table values are used to scale the quantized DCT image coefficients. This helps scale the coefficients back to normal values. Just as in the case of FDCT, a one to one mapping from the frequency to spatial domain samples is maintained by IDCT in retrieving all the original sample values. In principle, there is no loss in FDCT - IDCT process if proper precision is maintained.

The jpeg encoder is lossy at various stages. Firstly the DCT can lead to minor losses due to insufficient precision while dealing with cosine coefficients. This can be seen from the fact that when DCT is applied to component values and immediately IDCT is applied on them, the exact component values are not restored due to error in storage of the coefficients. Next, dequantization is a major source of loss in the entire procedure. During this process, the DCT coefficients get divided by fixed values taken from the quantization table. For higher qualities of compression, the quantization table contains values that tend to become 1 (so that there is no loss of information). For low qualities, the quantization table contains high values. Hence, most of the coefficients become zero and information is lost to some extent in the others.

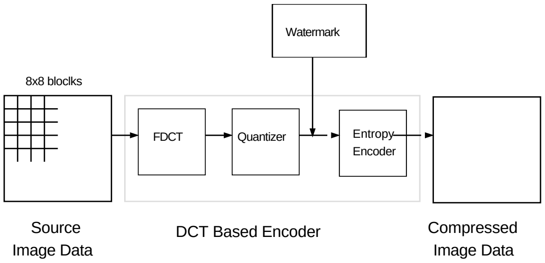


Figure 2: Watermark insertion during jpeg encoding

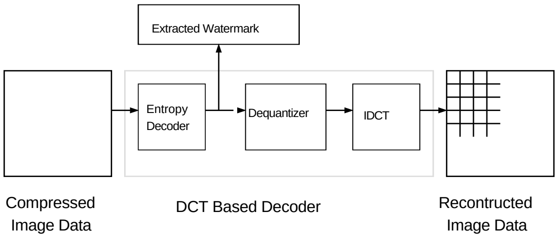


Figure 3: Watermark extraction during jpeg decoding

3.2.2 Method 1

Based on the above description, it is seen that one place to embed the desired information is in the quantized coefficients. This ensures that even the minute bit change will be retained. Figure 2 and 3 show the watermarking procedure. One way to do this is to use the LSB of the FDCT coefficients. We propose the following watermarking scheme: The image after being divided into 8x8 block undergoes the inter-component transform. Then DCT is applied. The obtained DCT coefficients undergo quantization where they get divided by values in the quantization table. It is in each of these that we embed our information. The last bit of the coefficients is replaced with an information bit. These coefficients are then entropy encoded. Figure 4 shows the order of embedding the watermark information in a typical 8x8 coefficient block. The gray colored squares indicate the coefficients selected for embedding the watermark. The proposed scheme uses the LSB of the quantized

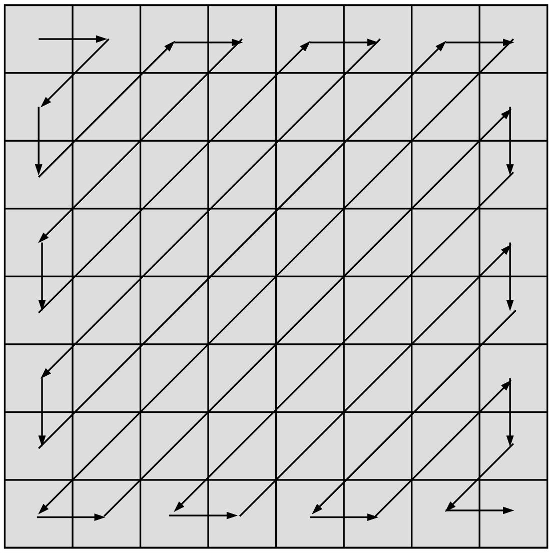
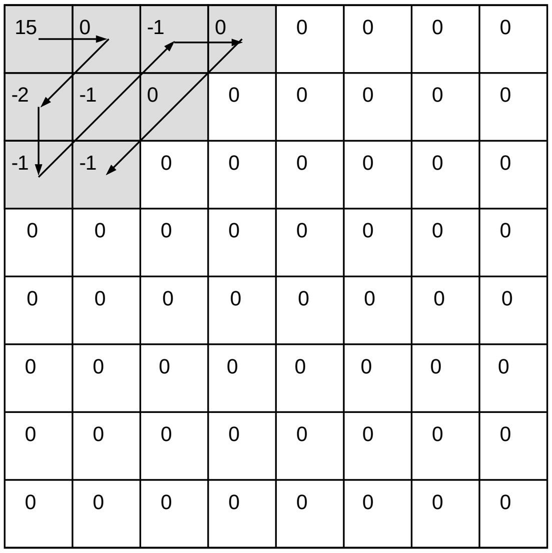


Figure 4: Order of embedding the information and the coefficients selected for embedding (shown in gray) when all the coefficients are considered



15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 5: Order of embedding of information and the coefficients selected for embedding (shown in gray) when only non-zero coefficients are considered.

coefficients for watermarking which has a drawback in the image decompression stage. When the image is compressed for high quality, the quantization table has small values. So the insertion of watermark information will introduce only small error at the decoding step. When the image is compressed for less quality, the quantization tables has large values. So the embedding of information (change of one bit) introduces a lot of error after it has been dequantized. As a result, the proposed scheme works well when compression at high quality is used but introduces visible distortion in the decompressed image when compression is used at low quality. This problem can be addressed as follows with the help of the decoder. Once it is known that the image has a watermark in it, after the quantized coefficients are obtained from entropy decoding, the data can be removed from the LSBs and set it to zero. The coefficients that were zero before embedding the watermark data turn back to zero. The high frequency coefficients that usually become zero in jpeg do not have the error introduced. The resulting decompressed image produced by this modification is quite acceptable. The proposed watermarking scheme has the following strengths: it has considerable amount of information storage capacity and information about a particular region can be kept in the region on an 8x8 block level. However, the weakness is an increase in the file size of the image. This is because many zero coefficients become non-zero when the watermark data is stored in the LSB of the coefficients. Since jpeg encodes all non- zero coefficients from the first to the last non-zero coefficients, this scheme increases the no of coefficients to be encoded.

3.2.3 Method 2

A second watermarking method we propose is to use only the non-zero quantized coefficients to embed the information. We embed watermark data in only the coefficients from the first non-zero coefficient to the last non-zero coefficient. The zero coefficients outside this range are not used which means no error is introduced in coefficients that are zero. The high frequency coefficients that usually become zero in jpeg do not have the error introduced. And how would decoder know which is the first and last coefficients that contain the information (now that the zero boundary is lost)? The solution is to use the first and the last non-zero coefficients as boundary markers and embed information between them. Thus, at the decoder end, while extracting the information from the array we ignore the first and the last non-zero coefficient. Figure 5 show the order of embedding the watermark information in a typical 8x8 coefficient block. The gray colored squares indicate the coefficients selected for embedding the watermark. The advantage of this method is that it does not increase the number of coefficients to be encoded. Hence there will be negligible change in the file size. It might even decrease since during the watermarking process, some coefficients that are non-zero might turn to zero. However, the tradeoff is that it has reduced information capacity because most of the coefficients become zero after quantization even when compressing at medium quality.

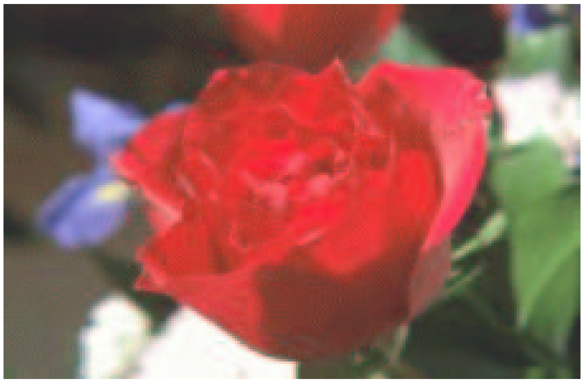


Figure 6: Original Image

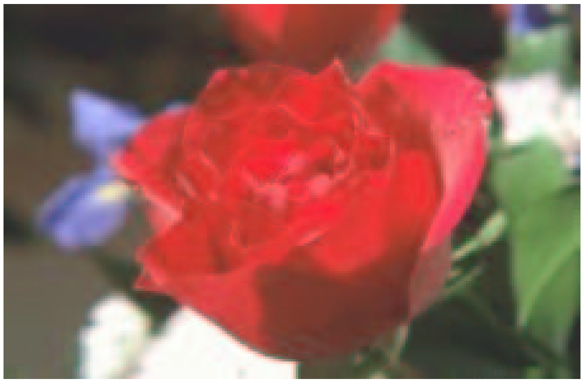


Figure 7: Original image at 90 quality factor

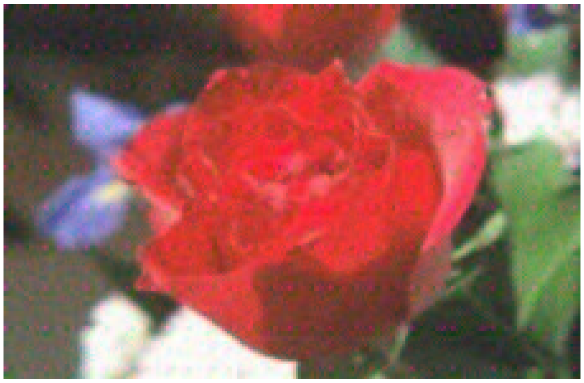


Figure 8: Watermarked image at 90 quality factor

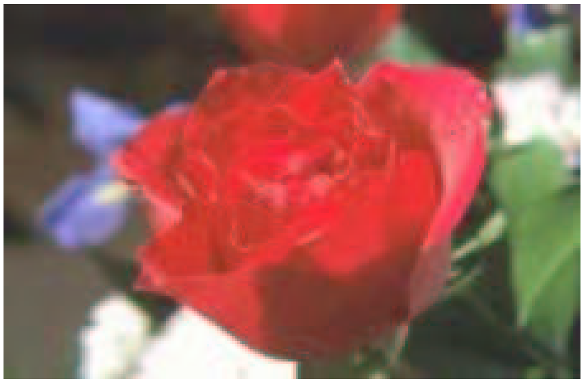


Figure 9: Original image at 50 quality factor

4. Results

In-order to carry out the watermark embedding in the middle of the jpeg encoding process, the free JPEG software from Independent JPEG Group has been used. This encoder has been modified to take watermark information as an input specification and embed the watermark after the quantization process as discussed. A similar approach is taken for the decoder. The decoder reads the watermark after the entropy decoding stage. It also outputs the watermark information apart from the decompressed image. Several images at various compression qualities have been tested. Figure 6 shows one of the images on which the experiments were carried out.

At high compression quality, the watermarking procedure does not produce noticeable distortion in the image. Figure 7 and 8 show the original image compressed at 90 quality factor and the corresponding watermarked image. The size of the original image at 90 quality factor is 10354 bytes. It was possible to embed as much as 6808 bytes of watermark information. The size of the compressed image after inserting the watermark at 90 quality factor is 19848 bytes.

However, when compression is at low quality, the error introduced in high frequencies is much larger and image is severely distorted. Figure 9 and 10 show the original image compressed at 90 quality factor and the corresponding watermarked image. The size of the original image at 50 quality factor is 4243 bytes. The watermark information is 6808 bytes. The size of the image after inserting the watermark at 50 quality factor is 16603 bytes.

In-order to remove the error introduced in the high frequencies at low qualities, the decoder is modified to remove the watermarked information from the image. Figure 11 shows the image at 50 quality factor and the decompressed image from which watermark has been removed during decoding. There is obviously no degradation in the decompressed image.

The second proposed watermarking method was also tested. Figure 12 shows the image at 50 quality factor in which only non-zero coefficients are chosen for watermarking. The amount of information that can be inserted is greatly reduced since most of the coefficents at 50 quality factor are zero and they are not used to embed infomation. The size of the watermarked image is 4147 bytes. (The size of the original image at 50 quality factor is 4243 bytes.)

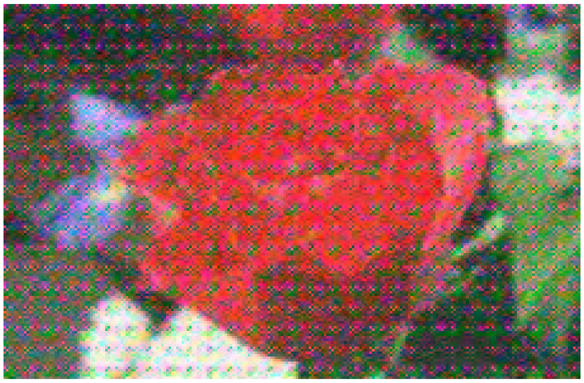


Figure 10: Watermarked image at 50 quality factor

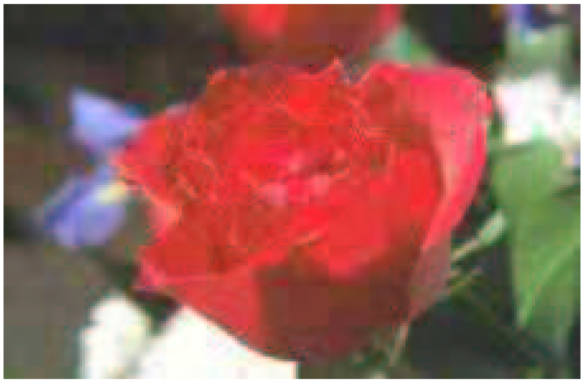


Figure 11: Watermark information removed with decoder help at 50 quality factor

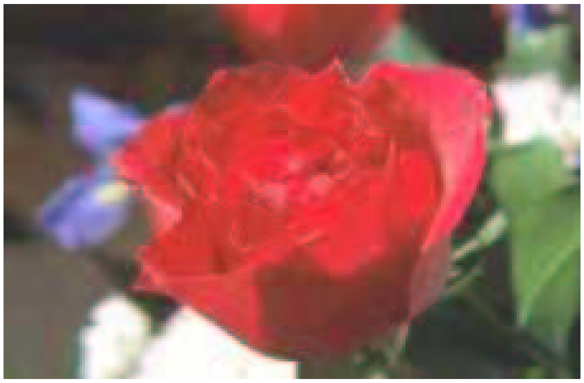


Figure 12: Image with watermark in only non-zero coefficients at 50 quality factor

5. Conclusion

Using watermarks for storing useful information is an attractive and useful proposition with many potential applications. We have examined the requirements on the watermarking methods for such purposes and found that some are conflicting. We have presented two methods that can effectively store good amount of watermarked information, robust to jpeg compression. The methods also demonstrate the existence of tradeoff between meeting different requirements on watermarking for carrying useful information.

References

[1] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[2] I. Cox, M. Miller, and J. Bloom. Watermarking applications and their properties. In *Proc. of Int. Conf. on Information Technology: Coding and Computing*, March 2000.

[3] R. J. A. Fabien A. P. Petitcolas and M. G. Kuhn. Information Hiding - A Survey. In *Proceegings of the IEEE, special issue on protection of multimedia content*, volume 87, pages 1062 – 1078, July 1999.

[4] S. Jain. Digital Watermarking Techniques: A Case Study in Finger Prints and Faces. In *Proceedings of the Indian conference on computer vision, graphics, and image processing (ICVGIP)*, 2000.

[5] J.-Y. C. Mercy George and N. Georganas. Spread Spectrum Spatial and Spectral Watermarking for Images and Video. In *Canadian Workshop on Information Theory*, 1999.

[6] N. Nikolaidis and I. Pitas. Digital image watermarking: An overview. In *ICMCS, Vol. 1*, pages 1–6, 1999.