Journal of Advances in Science and Technology

Vol. VI, Issue No. XII, February-2014, ISSN 2230-9659

A COMPARATIVE ANALYSIS ON VARIOUS STRATEGIES OF IMAGE COMPRESSION IN STEGANOGRAPHY

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

www.ignited.in

Journal of Advances in Science and Technology
Vol. VI, Issue No. XII, February-2014, ISSN 2230-9659

A Comparative Analysis on Various Strategies of Image Compression in Steganography

Soniya Wadhwa[1] Kiran Kumar[2]

[1]Research Scholar, Lingaya's University, Faridabad, Haryana

[2]Asst. Prof.

Abstract – Steganographic techniques can be used to hide data within digital images with little or no visible change in the perceived appearance of the image and can be exploited to export sensitive information. Since images are frequently compressed for storage or transmission, effective steganography must employ coding techniques to counter the errors caused by lossy compression algorithms. The Joint Photographic Expert Group (JPEG) compression algorithm, while producing only a small amount of visual distortion, introduces a relatively large number of errors in the bitmap data. It is shown that, despite errors caused by compression, information can be steganographically encoded into pixel data so that it is recoverable after JPEG processing, though not with perfect accuracy.

We demonstrate that one can adapt recent diffusion-based image compression techniques such that they become ideally suited for Steganographic applications. Thus, the goal is to embed secret images within arbitrary cover images. We hide only a small number of characteristic points of the secret in the cover image, while the remainder is reconstructed with edge-enhancing anisotropic diffusion in painting. Even when using significantly less than 1% of all pixels as characteristic points, sophisticated shapes of the secret can be clearly identified. Selecting more characteristic point's results in improved image quality. In contrast to most existing approaches, this even allows to embed large colour images into small grayscale images.

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the Internet. For hiding secret information in images, there exists a large variety of Steganographic techniques some are more complex than others and all of them have respective strong and weak points. Different applications have different requirements of the steganography technique used.

----------♦----------

INTRODUCTION

Two techniques are available to those wishing to transmit secrets using unprotected communications media. One is cryptography, where the secret is scrambled and can be reconstituted only by the holder of a key. When cryptography is used, the fact that the secret was transmitted is observable by anyone. The second method is steganography. Here the secret is encoded in another message in a manner such that, to the casual observer, it is unseen. Thus, the fact that the secret is being transmitted is also a secret.

Widespread use of digitized information in automated information systems has resulted in a renaissance for steganography. Information which provides the ideal vehicle for steganography is that which is stored with accuracy far greater than necessary for the data's use and display. Image, Postscript, and audio files are among those that fall into this category, while text, database, and executable code files do not.

It has been demonstrated that a significant amount of information can be concealed in bitmapped image files with little or no visible degradation of the image. This process, called steganography, is accomplished by replacing the least significant bits in the pixel bytes with the data to be hidden. Since the least significant pixel bits contribute very little to the overall appearance of the pixel, replacing these bits often has no perceptible effect on the image. To illustrate, consider a 24 bit pixel which uses 8 bits for each of the red, green, and blue color channels. The pixel is capable of representing 224 or 16,777,216 color values. If we use the lower 2 bits of each color channel to hide data, the maximum change in any pixel would be 26 or 64 color values; a minute fraction of the whole color space. This small change is invisible to the human eye. To continue the example, an image of 735 by 485 pixels could hold 735*485 * 6 bits/pixel * 1byte/8 bits = 267,356 bytes of data.

Kurak and McHugh show that it is even possible to embed one image inside another. Further, they assert that visual inspection of an image prior to its being downgraded is insufficient to prevent unauthorized flow of data from one security level to a lower one. A number of different formats are widely used to store imagery including BMP, TIFF, GIF, etc. Several of these image file formats "palletize" images by taking advantage of the fact that the color veracity of the image is not significantly degraded to the human observer by drastically reducing the total number of colors available. Instead of over 16 million possible colors, the color range is reduced and stored in a table. Each pixel, instead of containing a precise 24-bit color, stores an 8-bit index into the color table. This reduces the size of the bitmap by 2/3. When the image is processed for display by a viewer such as "xv", the indices stored at the location of each pixel are used to obtain the colors to be displayed from the color table. It has been demonstrated that steganography is ineffective when images are stored using this compression algorithm. Difficulty in designing a general-purpose Steganographic algorithm for palletized images results from the following factors: a change to a "pixel" results in a different index into the color table, which could result in a dramatically different color, changes in the color table can result in easily perceived changes to the image, and color maps vary from image to image with compression choices made as much for aesthetic reasons as for the efficiency of the compression.

In a time that is characterized by devices that allow anyone at any time to take and share high resolution digital images, the need for image compression codecs with high compression rates is more important than ever. A promising recent class of image compression codecs relies on in painting techniques based on partial differential equations (PDEs); see e.g.. In contrast to conventional methods such as JPEG, PDE-based image compression does not require transformations to the frequency domain: It simply stores a small fraction of characteristic pixels.

In the decoding step, the missing pixels are reconstructed by the in painting effect of the PDE. PDE-based approaches can outperform JPEG and even its sophisticated successor JPEG 2000. This observation holds all the more for perceptual quality metrics as shown in. However, they have not been adapted to specific application scenarios outside classical image compression so far.

IMAGE COMPRESSION

When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard Internet connection. In order to display an image in a reasonable amount of time, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyse and condense image data, resulting in smaller file sizes. This process is called compression.

In images there are two types of compression: lossy and lossless. Both methods save storage space, but the procedures that they implement differ. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate, resulting in close approximations of the original image, although not an exact duplicate. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group).

Lossless compression, on the other hand, never removes any information from the original image, but instead represents data in mathematical formulas. The original image's integrity is maintained and the decompressed image output is bit-by-bit identical to the original image input. The most popular image formats that use lossless compression is GIF (Graphical Interchange Format) and 8-bit BMP (a Microsoft Windows bitmap file).

Compression plays a very important role in choosing which Steganographic algorithm to use. Lossy compression techniques result in smaller image file sizes, but it increases the possibility that the embedded message may be partly lost due to the fact that excess image data will be removed. Lossless compression though, keeps the original digital image intact without the chance of lost, although is does not compress the image to such a small file size.

IMPLEMENTATION

The objectives of this work are to design and implement a Steganographic system which integrates a compression module and an encryption module to improve its capacity and security requirements. We discuss here the different phases involved in such implementation.

To embed secret data into a cover image requires two files. The first is the cover file or cover image that holds the hidden information. The second is the message – the information to be hidden. A message may be plain text, cipher text, other images or anything that can be embedded in a bit stream. When combined, the cover file and embedded message become a stego-file. A stego-key (a type of password) may be used to hide data, and then later used to decode the message.

Similarly to cryptography, secret key steganography requires the exchange of a secret key (stego-key) prior to communication. Secret Key Steganography takes a cover medium and embeds the secret message using the stego-key. Only the intended parties are able to reverse the process and read the secret message. In public key steganography, the sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message. The public key approach provides a more robust way of

implementing a Steganographic system because it can utilize a much more robust and researched technology. It has multiple levels of security in that unwanted parties must first suspect the use of steganography and then would have to find a way to crack the algorithm used by the public key system before they could intercept the secret message. Figure 1 shows the overall structure of the Steganographic system where fE denotes the Steganographic embedding function and fE -1 the Steganographic extracting function.

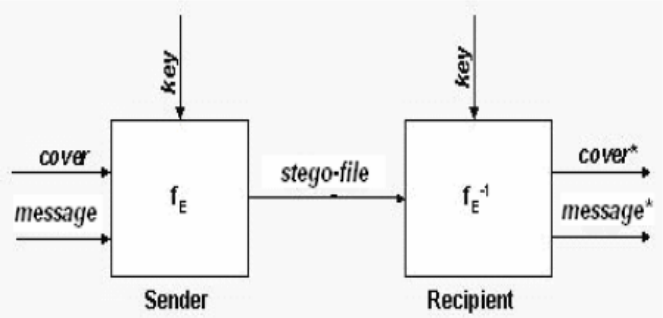

Figure 1. Overall structure of the Steganographic system

The Steganographic system allows the encoding and hiding of secret messages in cover images or audio files. The encryption and decryption algorithms implemented are the DES and the RSA algorithms using C# .NET libraries. Lossless and lossy compressions are also offered. The complete process which involves compression, encryption and hiding information is shown in Figure 2.

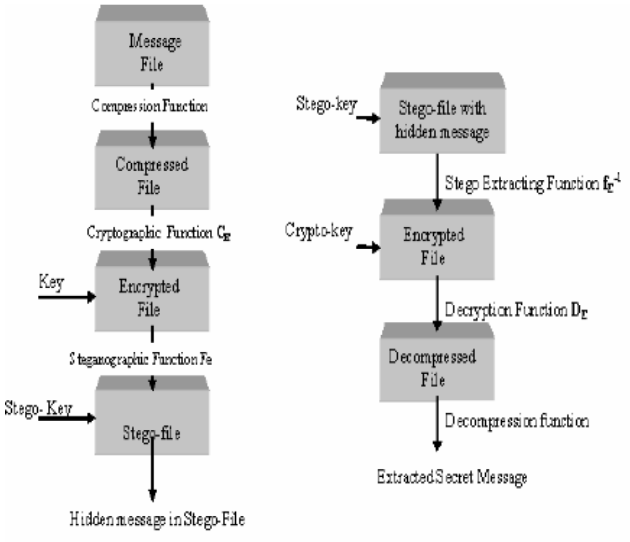

Figure 2. Embedding and Extracting Procedures

JPEG COMPRESSION

JPEG has been developed to provide efficient, flexible compression tools. JPEG has four modes of operation designed to support a variety of continuous-tone image applications. Most applications utilize the Baseline sequential coder/decoder which is very effective and is sufficient for many applications.

JPEG works in several steps. First the image pixels are transformed into a luminance/chrominance color space and then the chrominance component is down sampled to reduce the volume of data. This down sampling is possible because the human eye is much more sensitive to luminance changes than to chrominance changes. Next, the pixel values are grouped into 8x8 blocks which are transformed using the discrete cosine transform (DCT). The DCT yields an 8x8 frequency map which contains coefficients representing the average value in the block and successively higher-frequency changes within the block.

Each block then has its values divided by a quantization coefficient and the result rounded to an integer. This quantization is where most of the loss caused by JPEG occurs. Many of the coefficients representing higher frequencies are reduced to zero. This is acceptable since the higher frequency data that is lost will produce very little visually detectable change in the image. The reduced coefficients are then encoded using Huffman coding to further reduce the size of the data. This step is lossless. The final step in JPEG applications is to add header data giving parameters to be used by the decoder.

PDE-BASED IMAGE COMPRESSION

While anisotropic diffusion processes have been popular for denoising images for many years, their use in image compression is relatively new and a topic of ongoing research. Compared to conventional compression methods such as JPEG or JPEG 2000. diffusion-based image compression offers a number of advantages: It has a very intuitive physical interpretation, it is well-suited for extremely high compression rates, where it can outperform JPEG and even JPEG 2000 , and it can be used in a generic way for various types of data - including 2-D images, 3-D data sets, surface data and videos.

For simplicity, we consider a greyscale image, i.e. a function $f : \Omega \to \mathbb{R},$ where $\Omega \subset \mathbb{R}^2$ denotes a rectangular image domain. The extension to colour images does not create specific problems. The idea behind diffusion-based image compression is to store only the grey-values of a small subset $K \subset \Omega$. Using the pixels in K as Dirichlet boundary data, we reconstruct f in the unspecified domain $\Omega \setminus K$ as steady state $(t \to \infty)$ of the partial differential equation (PDE) $\partial_t u = \operatorname{div}(\boldsymbol{D}(\nabla u_\sigma)\nabla u)$, where $\nabla = (\partial_x, \partial_y)^\top$ is the

spatial nabla operator, and div the corresponding divergence operator. This PDE is known as edge-enhancing anisotropic diffusion (EED) . The diffusion process is steered by the positive definite 2x2 diffusion matrix D. It depends on the gradient of the Gaussian-smoothed version u_a of the image u, where <r is the standard deviation of the Gaussian. The direction of ∇u_σ is along the steepest ascent, i.e. perpendicular to image edges, and its magnitude $|\nabla u_\sigma|$ measures the strength of the edge (contrast). The two eigenvectors of Dare orthogonal resp. parallel to ∇u_σ, and their eigenvalues are given by $\mu_1 = 1$ hand $\mu_2 = \frac{1}{\sqrt{1+|\nabla u_\sigma|^2/\lambda^2}}$. Thus, along image edges, $\mu_1 = 1$ ensures that full diffusion is performed, while the decreasing behaviour of μ_2 w.r.t. $|\nabla u_\sigma|$ reduces diffusion across edges with high contrast. The parameter $\lambda > 0$ allows to steer this contrast dependence.

CONCLUSION

In this paper, we have developed a Windows based application implemented on the Microsoft .NET platform that allows a user to encrypt a secret message and hide its content into a cover image or audio file as well as extracting and retrieving the original data. Several compression, encryption and steganography techniques were combined and successfully implemented in order to provide a secure Steganographic system with different layers of security. Future work will include a broader variety of Steganographic techniques.

As an anti-steganography technique, JPEG is very effective. In order to achieve anything approaching an acceptable error rate, a great deal of redundancy must be applied. With the distance-to-origin and luminance modulo coding techniques the error rate can be brought down. But these techniques must be coupled with multiple redundancy to lower the error rate. Although the amount of data which can be hidden may be relatively small compared to the size of the image, the security threat that steganography poses cannot be completely eliminated by application of a transform-based lossy compression algorithm.

The success of these applications benefits heavily from the fact that diffusion-based image compression offers several properties that makes it preferable over classical codecs such as JPEG or JPEG 2000. This includes for instance its excellent performance for very high compression ratios, and its ability to exploit known boundary data in uncensoring applications. It appears that PDE-based compression and steganography constitute an ideal, hitherto unexplored match.

REFERENCES

- Artz, D., “Digital Steganography: Hiding Data within Data”, IEEE Internet Computing Journal, June 2001

- D.Artz, “Digital Steganography: Hiding data within data,” IEEE Internet Computing Vol 5, No. 3, pp. 75-80, May/June 2001.

- Galic, I., Zovko-Cihlar, B., Snje˘zana, R.D.: Computer image quality selection between JPEG, JPEG 2000 and PDE compression. In: Proc. 19th International Conference on Systems, Signals and Image Processing, IEEE Computer Society Press (2012) 437–441.

- Joint Photographic Experts Group (JPEG) Compression for the National Imagery Transmission Format Standard, MIL-STD-188-198A, December 1993.

- Mainberger, M., Bruhn, A.,Weickert, J., Forchhammer, S.: Edge-based image compression of cartoon-like images with homogeneous diffusion. Pattern Recognition 44 (2011) 1859–1873.

- N.Provos and P. Honeyman, “Hide and seek: An introduction to steganography,” IEEE Security & Privacy Vol 1, No. 3, pp. 32-44, May/June 2003.

- Pennebaker, William B., Mitchell, Joan L., JPEG Still Image Compression Standard, Van No strand Reinhold, New York, 1993.

- Sharp, T.: An implementation of key-based digital signal steganography. In Moskowitz, I.S., ed.: Information Hiding. Volume 2137 of Lecture Notes in Computer Science., Berlin, Springer (2001) 13–26.

- Su,P.C., Ku, C.C.J.: Steganography in JPEG2000 compressed images. IEEE Transactions on Consumer Electronics 49 (2003) 824 – 832

- Wallace, Gregory K., “The JPEG Still Picture Compression Standard”, Communications of the ACM, Vol. 34, No. 4, April 1991.