Prediction of Distortion Patterns in Image Steganography by Means of Fractal Computing

Shanyu Tang
Faculty of Computing
London Metropolitan University
London, UK
s.tang@londonmet.ac.uk

YongFeng Huang
Department of Electronic Engineering
Tsinghua University
Beijing, China
yfhuang@tsinghua.edu.cn

Abstract—**This paper describes a new method of predicting image distortion patterns in image steganography by using fractal computing. The method uses the successive random addition algorithm to simulate the image distortion patterns of embedding a secret image in an 'innocent' cover image, testing the fractal-like behaviour of image distortion patterns. The distortion patterns identified in the experimental data are characterised by a fractal Hurst parameter, which can be used to make predictions of future trends.**

Keywords-distortion patterns; image steganography; fractal computing

I. Introduction

Over the last three decades, people have sought ways to protect sensitive information against attack to make sure it is not received by unintended recipients. Conventional security measures are built on encryption, which encodes data such that an unintended recipient cannot determine its intended meaning. Encryption is now confronted with serious challenges since the increase in computational power has led to decryption of several classic encryption algorithms, indicating vulnerabilities in the encryption primitives. A major drawback to encryption is that the existence of data is not hidden. A solution to this problem is digital steganography [1].

Digital steganography is a form of data hiding in which a secret message is hidden within another file. It is essentially about embedding a message or file in another 'cover' file, called the carrier file, such that the carrier file is not altered enough to raise suspicion that something may be hidden within it [2]. Data to be hidden is the carrier medium; the cover file in which the data is hidden is the steganographic medium. Both parties communicating via steganography must use the same stego application.

Steganography in 'static' cover objects, such as plaintext, image files with BMP or JPEG format, and audio files in WAV or MP3 format, has been explored extensively [3] [4]. See [5] for a good survey of such techniques. There have been efforts to develop the steganalysis techniques for detecting steganography in static cover objects such as text, image or audio files. A larger number of image and audio steganalysis methods have been reported in the literature [6]-[9].

Previous image steganography studies have been largely focused on developing algorithms for steganography in image files and the steganalysis techniques, with no or little attention being given to image distortion patterns. In fact, image distortion is an intuitive indication of imperceptibility of image steganography. An understanding of image distortion patterns could help find ways to improving data embedding capacity, i.e. the number of bits in a byte of the cover image that can be replaced without affecting the image, which is a bottleneck to image steganography.

The use of data mining techniques and neural networks tools enables us to analyse digital information and understand future patterns, which may occur [10] [11]. As an example of applications, data mining techniques detect particular patterns in customer behaviour and future trends [12]. In this study, we attempt to use fractal computing methods to study whether image distortion patterns in steganography have burstiness behaviour - bursting on many or all scales. The burstiness behaviour is analogous to the self-similar or fractal-like behaviour, which exists in many natural phenomena such as Brownian motion, turbulent flow, atmospheric pressure, the distribution of stars and the activity of the stock market [13]-[15], which are much better characterised by fractal geometry theory than by Euclidean geometry.

The rest of this paper is organised as follows. Section II discusses fractal models, Section III describes the experimental setup for image steganography, Section IV presents the experimental results detailing distortions on the cover images of different sizes, as well as fractal modelling of image distortion patterns, and, finally, Section V concludes this work.

II. Fractal Computing

Fractal theory has been successfully used to depict many natural phenomena (for instance, random records in time) and more complex patterns associated with scientific phenomena [14], by characterising the structure embedded in one-, two- and three-dimensional space using fractal dimensions [15]. With the aid of fractal theory, some very

ISBN: 978-1-61208-158-8

complicated phenomena such as dendrites and chaos can be handled by fractal mathematical functions, which describe the dependence of different features on the various parameters (e.g., fractal dimensions). It is well established that fractal theory provides a basis for studying irregular sets by modelling and making predictions.

Applications of fractal modelling to Internet systems [16]-[18] led to the elucidation of the likely cause of web-based traffic that has implications for the performance and stability of web applications. Fractal mathematics was used to simulate the growth of biological systems [19][20]. Fractal methods were employed to gain an understanding of the behaviour of users in a multipoint, interactive communication scenario, particularly the dynamics of user participation at a session level [21].

Fractal random walk (Fractional Brownian motion) was proposed as a model for the TCP packet stream [17], although the FBM model greatly overestimated the loss probabilities in the operating regions of interest, and the assumptions of the model did not appear to be satisfied by the two data traces that were examined [16]. A mathematical investigation of the accuracy of this approximation was then needed. More recently, fractal statistical models have been using for predicting web data traffic patterns, and the preliminary experimental results are exciting [22].

Recently, fractal methods were used for modelling and coding of residuals for excitation in the linear predictive coding of speech [23]. The research shows fractal modelling reduced the bit-rate while maintaining quality; a 6 kbps speech coder was implemented using the piecewise self-affined fractal model. Fractal image compression techniques have been employed in the production of steganographic image files [24]. Fractal image compression encoded images at low bit-rate with acceptable image quality, but time taken for encoding was large. Muruganandham et al. proposed a fast fractal encoding using particle swarm optimisation (PSO) that optimises MSE between range block and domain block whereas preserving image quality [25].

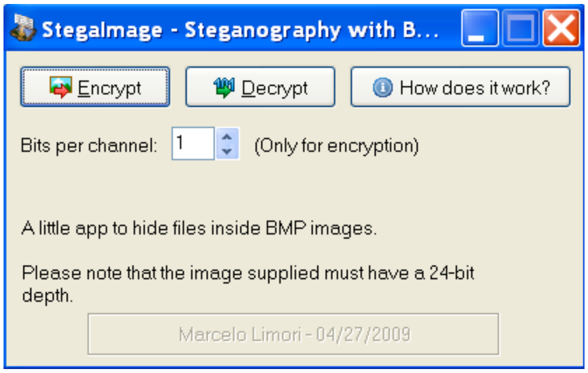
In this work, fractal models are suggested to elucidate the likely effects of the size of the cover image associated with different data embedding rates on image distortion patterns in image steganography.

III. Experimental Setup

Least significant bit (LSB) insertion is one of the methods of embedding secret information in an 'innocent' cover image. For instance, 10101001 is an 8-bit binary number. The last bit in the binary number is the least significant bit because changing it has the nominal effect on the value. The method is based on the fact that change in the LSB information of some area of the image will not be noticeable by the naked eye.

StegaImage is a type of computer software capable of hiding secret information within a cover image [26]. The cover image should be a bitmap image, in which the file to be hidden is embedded. The stego image, consisting of the cover image and the secret file, is indistinguishable from the original cover image. By using the LSB insertion method, the software embeds the secret file in a 24-bit bitmap image. Fig. 1 shows the interface of StegaImage.

It is worth mentioning the 'Encrypt' and 'Decrypt' buttons at the interface should be renamed as 'Data embedding' and 'Data extraction', respectively, so as to avoid any misunderstandings.



Bits per channel: 1 (Only for encryption)

A little app to hide files inside BMP images.

Please note that the image supplied must have a 24-bit depth.

Marcelo Limori - 04/27/2009

Figure 1. Steganographic software.

In a 24-bit bitmap, each pixel is represented as 3 bytes – one for red, one for green and one for blue, making up of a 24-bit number. Each colour is composed of 8-bit numbers, and the red, green and blue colours/channels create the final colour of the pixel. To hide something inside the image, software will replace the LSB of each 8-bit channel of every pixel, with the bits of the file to be hidden. This means the last bit in a byte can be overwritten without affecting the colour it appears to be. Therefore, the size of the cover image must be bigger than the size of the image to be hidden in order to accommodate in unrecognisable manner.

IV. Results And Discussion

Three bitmap images with different sizes were used as cover objects in the experiments to hide smaller images, respectively. The sizes of the cover images vary from 772 KB, 1.34 MB to 1.56 MB.

A. Steganography in a 772 KB CoverImage

Fig. 2 shows the image to be hidden, the cover image (772 KB) and the stego images at various numbers of bits inserted into each channel (i.e. each pixel consisting of 24 bits).

A map image was used as the image to be hidden, which was embedded in the 772 KB bitmap image using StegaImage. No distortion was noticeable on the stego image when the map image was hidden in the cover image at 1 bit insertion per channel. However, inserting 5 or more bits per channel led to significant distortion to the original cover image.

In view of LSB data embedding, the cover image has the maximum data embedding capacity, depending on the number of colours and palette in the cover image. With StegaImage, the bits of the image to be hidden were spread all over the least significant bits of the pixels of the cover image under a certain threshold of bits insertion per channel (here 5 bits per channel); when the number of bits inserted into a pixel was greater than the threshold, the distortion appeared on the upper side of the cover image due to the design principle of the software [26].

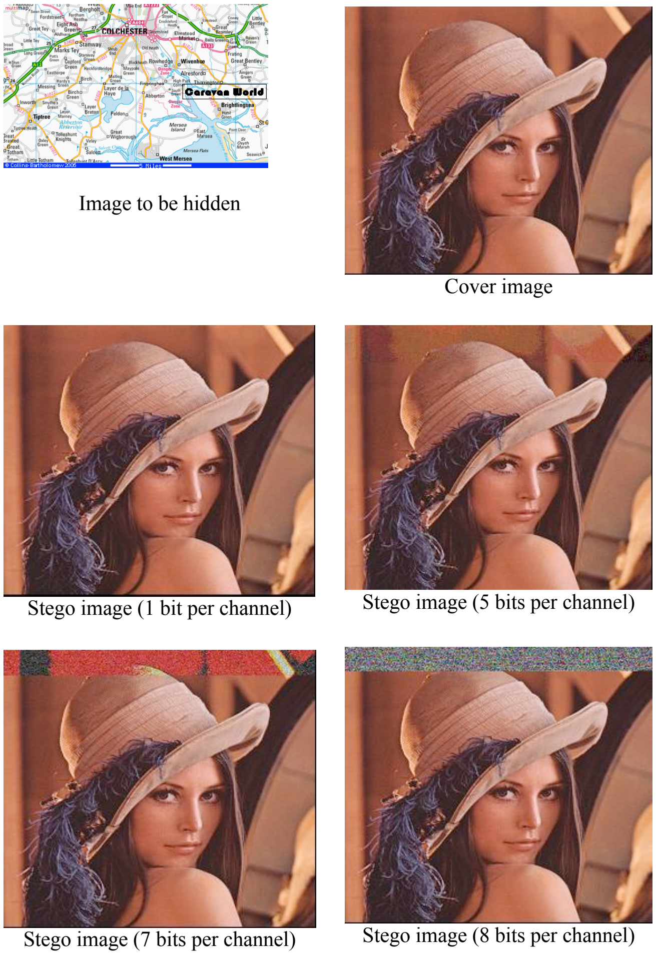


Image to be hidden

Cover image

Stego image (1 bit per channel)

Stego image (5 bits per channel)

Stego image (7 bits per channel)

Stego image (8 bits per channel)

Figure 2. Steganography in a 772 KB cover image.

B. *Steganography in a 1.34 MB Cover Image*

Fig. 3 shows the map image to be hidden, the cover image (1.34 MB) and the stego images, each containing the cover image and the map image, as well as a distortion on the cover image when the map image was hidden in the cover image at up to 8 bit insertion per channel.

The StegaImage software uses last bits of each channel for insertion of data. If only the LSB was changed there was no effect on the cover image after data embedding; however, when the number of insertion bits per channel increased to 7, the change in the original cover image was perceptive, as shown on the last two stego images in Fig. 3.

C. *Steganography in a 1.56 MB Cover Image*

Fig. 4 shows the hidden image, the cover image (1.56 MB) and the stego images at various numbers of bits inserted into each channel using StegaImage software.

StegaImage was used to hide the picture of a girl inside the picture of a sport car. No difference between the cover image and the stego image could be seen by the naked eye if 5 bits per channel were replaced with the bits of the image to be hidden. But when the number of replaced bits per channel increased to 6, image distortion was clearly noticeable on the upper side of the cover image when compared with the original cover image.

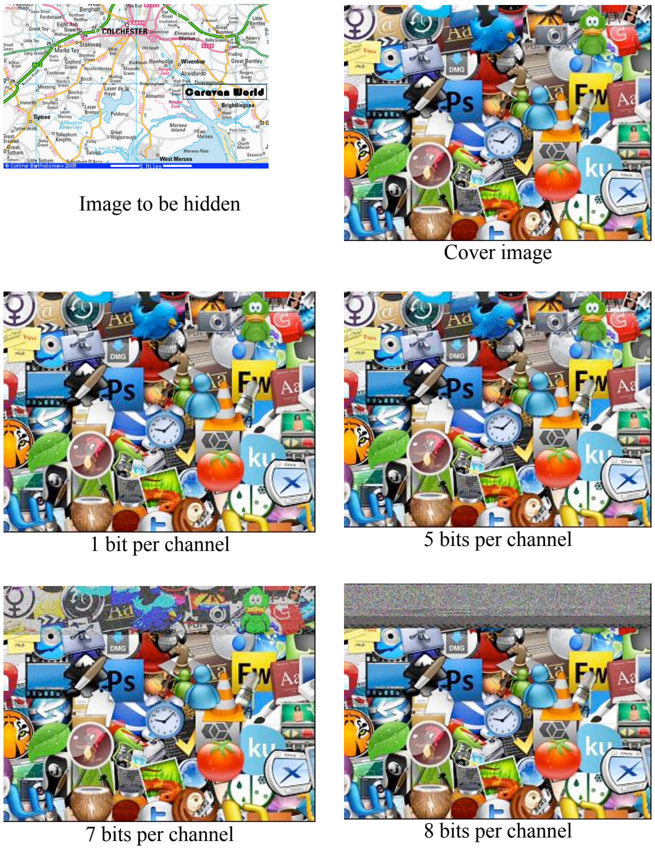


Image to be hidden

Cover image

1 bit per channel

5 bits per channel

7 bits per channel

8 bits per channel

Figure 3. Steganography in a 1.34 MB cover image.

D. *Effect of the size of the image to be hidden*

The StegaImage software that was used in our experiments to perform image steganography is based on LSB substitution, that is, the last bit in the binary number of the cover image is replaced with one bit of the image to be hidden. Such change in the LSB information of some area of the cover image will not be noticeable by the naked eye. So the size of the image to be hidden must be smaller than the size of the cover image.

Our on-going research has been taking into account the size of the image to be hidden. It is anticipated that if the image to be hidden is very small in relation to the cover image, the hiding process may not produce a noticeable distortion in the cover image, depending on the ratio of the size of the image to be hidden to the size of the cover image, as well as the number of colours of the cover image.

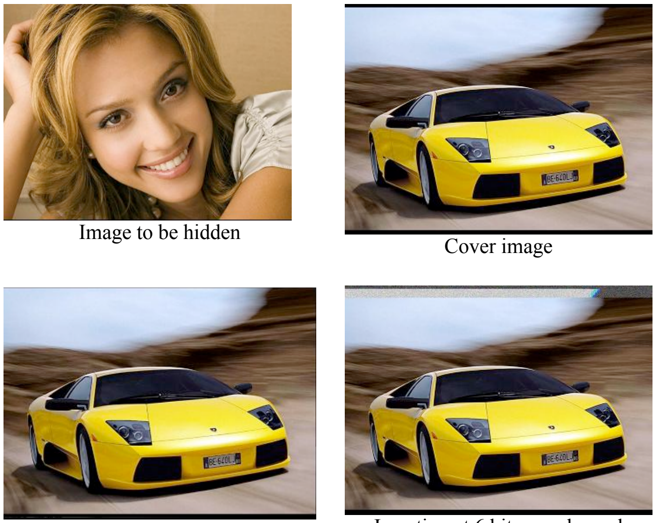


Image to be hidden

Cover image

Insertion at 5 bits per channel

Insertion at 6 bits per channel

Figure 4. Steganography in a 1.56 MB cover image.

E. *Fractal Modelling of Image Distortion Patterns*

The maximum number of bits per channel that can be replaced without causing image distortion is a measure of the threshold of image distortion, which can be used to estimate the data embedding capacity. Table I lists the thresholds of image distortion for the three steganography experiments detailed in the previous sections.

TABLE I. THRESHOLD OF IMAGE DISTORTION

Size of cover image	Threshold
772 KB	5 bits / channel
1.34 MB	7 bits / channel
1.56 MB	6 bits / channel

To simulate the relationship between the threshold of image distortion and the size of the cover image, we used the successive random addition algorithm introduced by Voss [27] to generate fractal random walk simulation.

Fractal random walk simulation has been used for characterising random fractals [14], which may seem analogous to the threshold of image distortion vs. the size of the cover image, as randomness is inherent in most phenomena. A sequence of increments for random walk, i.e. increase in the threshold of image distortion in image steganography, can be generated from a sequence of Gaussian random variables, and the normalised variance of increments (V) is a function of the parameter t [15], $V(t) = |t|^{2H}$, where H is the Hurst parameter. Thus, the plot of $V(t)$ against t on a log-log plot has a slope, which is an estimation of $2H$.

In this case, V(z) represents the fluctuation of the threshold of image distortion in image steganography and is a function of the size of the cover image (z), given by

$$V(z) \propto |z|^{2H} \qquad (1)$$

which is fitted with a range of statistic equations, and z is the size of the cover image used to embed the image to be hidden. The value of the Hurst parameter H will characterise the degree of burstiness of image distortion.

Fig. 5 shows the threshold of image distortion as a function of the size of the cover image based on the steganography experimental results listed in Table I. A fractal Hurst parameter of H = 3.1973 / 2 = 1.5987 was obtained from the slope of log (Threshold) vs. log (Size of cover image) according to (1).

According to random fractal theory, the value of the Hurst parameter should be between 1 and 2. Giving that the experimental value of H is equal to 1.5987, it indicates that image distortion has fractal-like behaviour in terms of the threshold of bits insertion per channel as a function of the size of the cover image. The Hurst parameter would maintain on many or all scales, so its value could be then used to describe the dependence of the image distortion threshold on the size of the cover image, even for a very large cover image, e.g., 100MB in size.

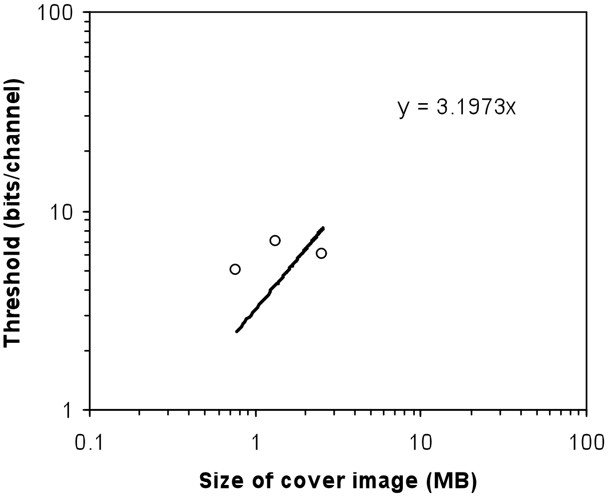


Figure 5. Threshold of image distortion as a function of the size of the cover image.

V. Conclusion And Future Works

In this paper, we have applied fractal computing to modelling of image distortion patterns in image steganography. The successive random addition algorithm has been used to simulate the image distortion patterns of embedding the image to be hidden in the cover image. A distortion pattern is beginning to emerge from our analysis of the steganography experimental data. The distortion pattern identified in this work could make predictions of future trends.

Other fractal methods such as R/S scaling and Fractal random walk could be used to simulate the image distortion patterns of image steganography in the future. Comparisons in the fractal Hurst parameters obtained from different fractal models would be able to further confirm the fractal behaviour of image distortion patterns in image steganography. In addition, investigation into the effect of the size of the image to be hidden on image distortion patterns will also be the subject of future work.

Acknowledgment

The authors would like to thank M.K.M. Jayawardena, G. S. Gayan De Fonseka, and B. Abhishek for providing some images, and thank Marcelo Limori for developing the StegaImage steganographic tool.

This work was supported in part by the British Government (Grant ktp8263).

References

[1] D. Artz, “Digital Steganography: Hiding data within data,” IEEE Internet Computing, pp. 75-80, June 2001.

[2] J. E. Wingate, “Digital Steganography: An introduction to the practice of digital information hiding,” Digital Forensics Magazine, pp. 73-76, June 2010.

[3] F. A. P. Peticolas, R. J. Anderson, and M. G. Kuhn, “Information Hiding – A Survey,” IEEE Trans. Proc. Thy., vol. 87, no. 7, pp. 1062-1078, 1999.

[4] P. Bao, and X. Ma, “MP3-resistant music steganography based on dynamic range transform,” Proc. IEEE International Symposium on Intelligent Signal Processing and Communication Systems, Seoul, Korea, Nov. 2004, pp. 266-271.

[5] N. F. Johnson, Z. Duric, and S. Jajodia, Information hiding: Steganography and watermarking-attacks and countermeasures, Kluwer Acade. Publishers, 2000.

[6] J. Fridrich, M. Goljan, and R. Du, “Detecting LSB Steganography in Colour and Grayscale Images,” IEEE Multimedia, vol. 8, pp. 22-28, 2001.

[7] S. Dumitrescu, X. Wu, and Z. Wang, “Detection of LSB Steganography via Sample Pair Analysis,” IEEE Transactions on Signal Processing, vol. 51, no. 7, pp. 1995-2007, 2003.

[8] S. Lyu, and H. Farid, “Detecting hidden messages using higher-order statistics and support vector machines,” Lecture Notes in Computer Science, vol. 2578, pp. 340-354, 2003.

[9] Y. Miche, B. Roue, A. Lendasse, and P. Bas, “A feature selection methodology for steganalysis,” Proc. International Workshop on Multimedia Content Representation, Classification and Security, Istanbul, Turkey, September 2006, pp. 49-56.

[10] L. D. Olsen, and D. Delen, Advanced Data Mining Techniques, Springer, USA, 2008.

[11] S. Ahmed, P. Pan, and S. Tang, “Clustering websites using a MapReduce programming model,” Journal of Communication and Computer, USA, vol. 7, no. 9, pp. 18-26, 2010.

[12] R. S. Ahmed, “Applications of Data Mining in Retail Business,” in Information Technology: Coding and Computing, vol. 2, 2004, pp. 455-459.

[13] J. Feder, Fractals, Plenum Press, New York, 1988.

[14] K. Falconer, Fractal Geometry: Mathematical Foundations and Applications, John Wiley & Sons, Chichester, 1990, pp.146-160.

[15] B. H. Kaye, A Random Walk Through Fractal Dimensions, VCH, Weinheim, 1994, pp. 179-188.

[16] D. Heyamn, “Some issues in performance modelling of data teletraffic,” Performance Evaluation, vol. 34, pp. 227-247, 1998.

[17] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the self-similar nature of Ethernet traffic,” IEEE/ACM Trans. Networking, vol. 2, pp. 1-15, 1994.

[18] K. Rezaul, S. Tang, T. Wang, and A. Paksta, “Empirical distribution function test for world wide web traffic,” Proc. IADIS International Conference Applied Computing 2004, Lisbon, Portugal, March 2004.

[19] S. Tang, Y. Ma, and I.M. Sebastine, “The fractal nature of Escherichia coli biological flocs,” Biointerfaces, vol. 20, pp. 211-218, 2001.

[20] S. Tang, “Computer simulation of fractal structure of flocs,” Encyclopaedia of Surface and Colloid Science, pp. 1162-1168, August 2006. Taylor & Francis, ISBN: 978-0-8493-9615-1.

[21] S. Bhatti, “Modelling user behaviour in networked games,” Proc. the ninth ACM international conference on Multimedia, Ottawa, Canada, 2001, pp. 212 - 220.

[22] S. Tang, and H. Kazemian, “Simulation of web data traffic patterns using fractal statistical modelling,” Lecture Notes in Computer Science, Springer, in press, 2011.

[23] W. Kinsner, and E. Vera, “Fractal modelling of residues in linear predictive coding of speech,” Proc. 8th IEEE International Conference on Cognitive Informatics, 2009, pp. 181-187.

[24] P. Davern, and M. Scott, “Fractal based image steganography,” Lecture Notes in Computer Science, vol. 1174, pp. 279-294, 1996.

[25] A. Muruganandham, and R. S. D. Wahida Banu, “Effective MSE optimization in fractal image compression,” International Journal of Computer Science and Information Security, vol. 8, no. 2, 2010.

[26] Computerworld. Steganography: Hidden Data. Quick study by Deborah Radcliff. [online] 2010. Available at http://www.computerworld.com/securitytopics/security/story/0,10801,71726,00.html.

[27] R. F. Voss, “Random fractal forgeries,” in Fundamental Algorithms for Computer Graphics, vol. 17, R. A. Earnshaw, Eds, NATO ASI Series F, Computer and System Sciences, 1985.