Robust Steganography Using Texture Synthesis

Zhenxing Qian[1], Hang Zhou[2], Weiming Zhang[2], Xinpeng Zhang[1]

1. School of Communication and Information Engineering, Shanghai University, Shanghai, 200444, China
2. School of Information Science and Technology, University of Science and Technology of China, Hefei, 230026, China
Email: zxqian@shu.edu.cn

Abstract. This paper proposes a robust steganography based on texture synthesis. Different from the traditional steganography by modifying an existing image, we hide secret messages during the process of synthesizing a texture image. The generated stego texture is similar to the sample image, preserving a good local appearance. Large embedding capacities can be achieved proportional to the size of the synthesized texture image. This algorithm also ensures that the hidden message can be exactly extracted from the stego image. Most importantly, the proposed steganography approach provides a capability of countering JPEG compression.

Keywords: Steganography, information hiding, texture synthesis

1 Introduction

Steganography is a technology of covert communication, which hides secret information into a cover media so as to avoid the eavesdropper's suspicious [1]. Nowadays, secret messages are always embedded into the digital media such as digital image, video, audio, text, etc. Many steganography methods have been developed in the past few decades [7-9].

Traditionally, secret messages are hidden by overwriting the insignificant data of a chosen cover. Given a multimedia, the embedding capacity is determined by the allowed distortions. More distortions would result in more risk of defeat by an eavesdropper using steganalysis tools. Another emerging problem is that the transmission is not always lossless. For example, when uploading an image onto the social network, this image is always compressed by the service provider. Since the feature of robustness is not considered in traditional steganography methods, message extraction in most of the algorithms would fail when the stego is processed. Hence, both the large embedding capacity and the robustness are required in modern steganography.

To this end, this paper proposes a novel steganography method based on texture synthesis. Both the capacity and robustness can be achieved. With a small texture pattern, we construct a message-oriented texture image with proportional size to accommodate message with arbitrary length. The generated stego texture image is ro-

J.-S. Pan et al. (eds.), *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, Smart Innovation, Systems and Technologies 63,
DOI 10.1007/978-3-319-50209-0_4

bust to JPEG compression, i.e., hidden massage can still be correctly extracted from the compressed stego image.

Until now, few steganography works based on texture synthesis have been done. Pioneering works were done by Otori and Kuriyama [2][3]. Secret messages are regularly arranged into colored dotted pattern using the colors picked from a texture sample with features corresponding to the embedded data. These dotted patterns are written onto a blank canvas, the blank regions of which are synthesized using the texture samples. Embedding capacity of the method is determined by the dotted patterns painted on the image. Another synthesis based steganography method was investigated by Wu and Wang [4]. A texture synthesis process re-samples a smaller texture image to construct a new texture image. Secret messages are concealed through the process of texture synthesis, in which sorted candidate patches are mapped to secret bits. During data extraction, they first recover the original texture image, and then extract the hidden message using the reordered candidate patches. This method can achieve high embedding rate than the works in [2] and [3].

In these steganography methods in [2]~[4], the constructed stego texture images have good appearance and the embedding rates are considerable. However, once the stego image is compressed by the tools like JPEG, many errors would happen during messages extraction. To overcome this problem, we propose a new synthesis based steganography approach, in which both large capacity and robustness can be realized.

2 Proposed Method

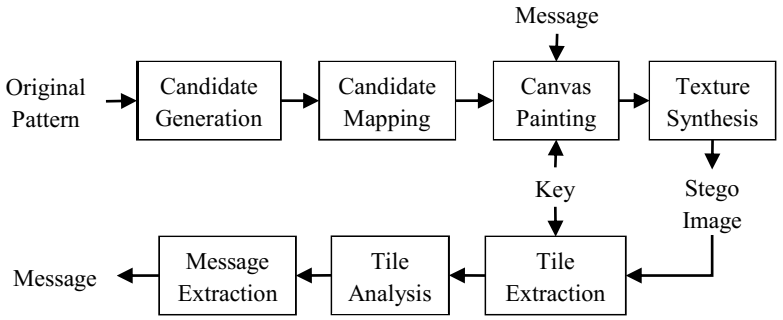


Fig. 1 Framework of the proposed method

Framework of the proposed method is shown in Fig. 1. To hide a secret message, a data hider uses a source texture pattern and divides this patter into overlapped candidate tiles. These candidates are mapped to several classifications by analyzing their texture complexities. With a secret key, the data hider pseudo-randomly distributes the selected candidate tiles to a blank canvas. Each tile is selected according to a segment of message. Other regions in blank are then synthesized by choosing the best tile from all candidates. On the receiver side, the secret key is used to identify the tiles containing secret bits. By analyzing texture of all the extracted tiles, several categories are reconstructed. Accordingly, the hidden bits can be extracted from each tile.

2.1 Data Hiding

Given a source texture pattern with the size of $S_r \times S_c$, we first divide it into a number of candidate tiles. The division is done by shifting each pixel following the raster-scan order, resulting in $(S_r-T_r+1)\cdot(S_c-T_c+1)$ overlapped tiles. Each tile is sized $T_r \times T_c$. We further divide each tile into the kernel area and boundary area, as shown in Fig. 2. Each kernel contains $K_r \times K_c$ pixels. The boundary depths in both directions are B_r and B_c, respectively. Therefore, there are $(S_r-T_r+1)\cdot(S_c-T_c+1)$ kernels corresponding to all the candidate tiles.

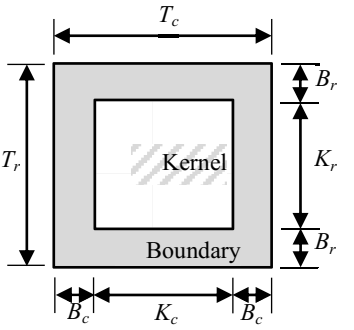


Fig. 2 A tile containing the kernel area and the boundary area

Denote all the tiles as $\{\mathbf{P}_1,\mathbf{P}_2,\ldots,\mathbf{P}_N\}$ and corresponding kernels as $\{\mathbf{R}_1,\mathbf{R}_2,\ldots,\mathbf{R}_N\}$, where $N=(S_r-T_r+1)\cdot(S_c-T_c+1)$. We evaluate the complexity degree of each kernel by standard deviation using

$$D_k = \left[\left\{ \frac{1}{K_r K_c - 1} \sum_{i=1}^{K_r} \sum_{j=1}^{K_c} \left(R_k(i,j) - A_k \right)^2 \right\}^{1/2} \right] \tag{1}$$

where $[\cdot]$ is the rounding operator, $k=1,2,\ldots,N$, and

$$A_k = \sum_{i=1}^{K_r} \sum_{j=1}^{K_c} R_k(i,j) \Big/ K_r K_c \tag{2}$$

Denote that the degree values $\{D_1,D_2,\ldots,D_N\}$ range from $D_{\min}$ to $D_{\max}$. Accordingly, we compute M values $\{V_1,V_2,\ldots,V_M\}$, in which

$$V_i = \left[\frac{D_{\max} - D_{\min}}{M} (i-1) + \frac{D_{\max} - D_{\min}}{2M} \right] \tag{3}$$

With these values, we construct M categories $\{\mathbf{\Pi}_1, \mathbf{\Pi}_2,\ldots, \mathbf{\Pi}_M\}$ containing candidate tiles by

$$\mathbf{\Pi}_i = \left\{ \mathbf{P}_k \mid D_k \in (V_i - \delta, V_i + \delta) \right\} \tag{4}$$

where i=1,2,…,M, k=1,2,…,N, and δ is a smaller number satisfying $0 \le \delta < (D_{\max} - D_{\min})/2M$.

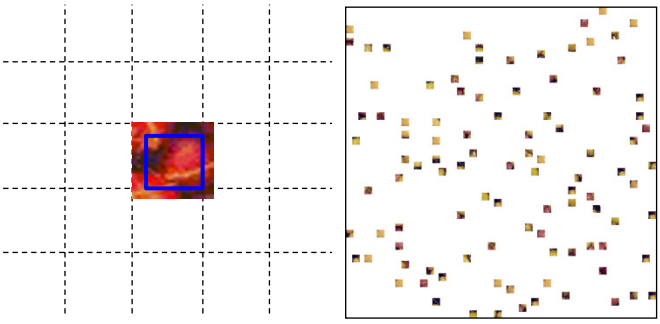
We use each category to represent several secret bits. Since there are M categories, each one stands for $\lfloor \log_2 M \rfloor$ bits, where $\lfloor \cdot \rfloor$ the rounding down operator. For example, if M=4, $\mathbf{\Pi}_1$, $\mathbf{\Pi}_2$, $\mathbf{\Pi}_3$, and $\mathbf{\Pi}_4$ stand for the secret bits of "00", "01", "10" and "11", respectively.

To hide a secret message, we turn the message into binary bits and divide these bits into segments with each containing $\lfloor \log_2 M \rfloor$ bits. Assuming L such segments $\{\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_L\}$ are included in the message, we calculate the decimal values E_i (i=1,2,…,L) for all these segments. Construct a blank canvas with the size of $[(K_r+B_r)\cdot W_r] \times [(K_c+B_c)\cdot W_c]$, where W_r and W_c are integers satisfying

$$W_r \cdot W_c > 4L / \lfloor \log_2 M \rfloor .$$

With a secret key, we pseudo-randomly generate L integer pairs $\{(p_1,q_1),(p_2,q_2),\ldots,(p_L,q_L)\}$, in which $1 \le p_i \le W_r$ and $1 \le q_i \le W_c$. Two conditions are required during the generation. First, each pair should be different from any other one. Second, for arbitrary two pairs (p_i,q_i) and (p_j,q_j), either $p_i - p_j > 1$ or $q_i - q_j > 1$ should be satisfied, where i=1,2,…,L and j=1,2,…,L.

For each segment $\mathbf{B}_i$ (i=1,2,…,L), we arbitrarily choose one tile from the category $\mathbf{\Pi}_{E_i+1}$. Then we paint all $T_r \cdot T_c$ pixels in this tile onto the canvas from the pixel at $((K_r+B_r)\cdot(p_i-1)+1, (K_c+B_c)\cdot(q_i-1)+1)$ to the pixel at $((K_r+B_r)\cdot p_i+B_r, (K_c+B_c)\cdot q_i+B_c)$. Fig. 3(a) shows the diagram of painting one tile onto the canvas, in which the content surrounded by the blue square is the kernel of the candidate tile. Fig. 3(b) shows an example of painting 100 tiles containing 200 secret bits onto a blank canvas.



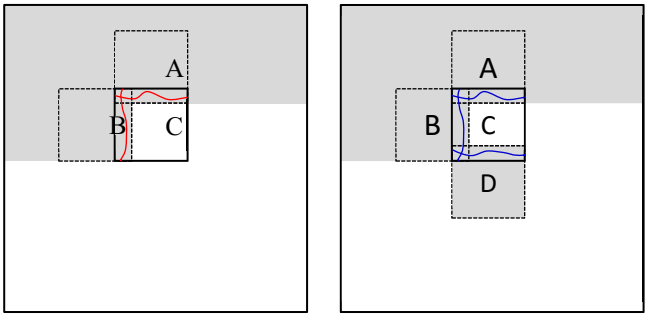
(a) (b)

Fig. 3 Painting selected tiles onto a blank canvas. (a) Diagram of painting one tile onto a blank canvas. (b) An example of painting 100 tiles containing 200 secret bits onto a canvas.

The other blank regions on the canvas are then painted by texture synthesis. From all $(S_r - T_r + 1)\cdot(S_c - T_c + 1)$ candidates, proper tiles are identified. We slightly modify the "image quilting" algorithm proposed by Efros and Freeman in [5]. In [5], synthesis is realized by iteratively padding chosen identical sized candidates to a blank window. Since there are overlapped regions, errors between the chosen block and the existing blocks at the overlapped region are computed. Generally, a best tile that has the smallest mean square errors (MSE) of the overlapped parts is selected. A diagram is

illustrated in Fig. 4(a). The regions in gray color stand for the synthesized contents. When synthesizing the content of "**C**", MSE of the overlapping regions between "**C**" and the *upper* tile "**A**", and MSE between "**C**" and the *left* tile "**B**" are calculated. One candidate that has the smallest MSE is chosen as the best. Then, the minimum cost path along the overlapped surface is computed to find the seam, see the red curves on the overlapped region, and the content is pasted onto the canvas along the seams. Details of the algorithm can be found in [5].

In the proposed method, the main difference is that many tiles containing secret bits have been painted on the canvas. We slightly modify the quilting algorithm in [5] to construct the synthesized image with good appearance. As aforementioned, the key controlled painting ensures that no blocks containing secret bits are adjacent. Instead of calculating MSE of *upper* and *left* overlapping parts, the *right* or *down* overlapping parts are also included to identify the best candidate. An example is shown in Fig. 4(b), in which "**D**" is the painted tile containing secret bits. We find the best candidate for "**C**" by computing MSE from three directions, and find the best seams for the overlapped surfaces, see the blue curves on overlapped regions.



(a) (b)

Fig. 4. Synthesizing the blank regions. (a) Diagram of the algorithm proposed by Efros and Freeman. (b) Diagram of the modified algorithm in the proposed method.

After painting all the blank regions of the canvas, a steganography image is finally generated. The embedding capacity C_e (bits) of the proposed algorithm is thus equal to

$$C_e = L \cdot \lfloor \log_2 M \rfloor \tag{5}$$

2.2 Data Extraction

On the recipient end, the secret key is used to identify the positions of tiles containing secret bits. The receiver extracts L tiles $\{\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_L\}$ from the stego image. Correspondingly, L kernels $\{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_L\}$ are separated from these tiles using the same way as illustrated in Fig. 2. Complexities of these kernels are calculated by (1) and (2). Denote these degree values as $\{G_1, G_2, \ldots, G_N\}$ range from $G_{\min}$ to $G_{\max}$.

Since the stego image may be compressed by JPEG, the calculated degree values might not belong to the original degree set $\{V_1,V_2,\ldots,V_M\}$. The receiver reconstructs a new set $\{V_1',V_2',\ldots,V_M'\}$ by

$$V_i' = \left\lfloor \frac{G_{\max}-G_{\min}}{M}(i-1) + \frac{G_{\max}-G_{\min}}{2M} \right\rfloor \tag{6}$$

With these values, the tiles are re-classified into M new categories $\{\mathbf{\Lambda}_1,\mathbf{\Lambda}_2,\ldots,\mathbf{\Lambda}_M\}$ containing candidate tiles by

$$\mathbf{\Lambda}_i = \{\mathbf{Q}_k \mid V_k' \in [V_i'-\Delta/2, V_i'+\Delta/2)\} \tag{7}$$

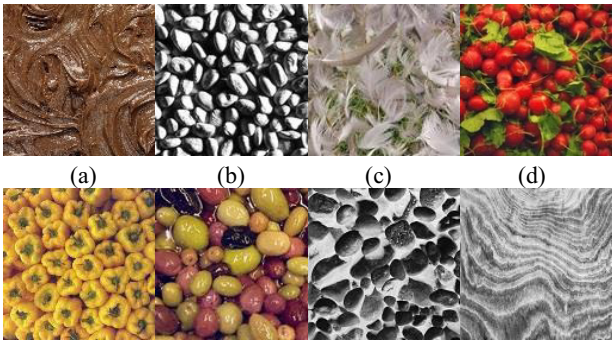
where i=1,2,…,M, k=1,2,…,L, and

$$\Delta=(G_{\max}-G_{\min})/M.$$

From the each extracted tile $\mathbf{Q}_k$ (k=1,2,…,L), $\lfloor \log_2 M \rfloor$ hidden bits can extracted. If $\mathbf{Q}_k \in \mathbf{\Lambda}_i$ (i=1,2,…,M), the hidden bits would be the binary bits of (i−1). This way, $L\cdot\lfloor \log_2 M \rfloor$ bits of hidden bits can be extracted from the received image.

3 Experimental Results

The proposed algorithm is carried out in many images. A group of image is shown in Fig. 5. The source patches sized 128×128 are shown in Fig. 5(a)~(h), containing several kinds of textures. Fig. 5(i)~(p) are the generated steganography images with the sizes of 653×653, corresponding to 2500 overlapped tiles, *i.e.*, W_r=50 and W_c=50. All of the experiments use a fixed δ as 0. The main parameters are set as M=16 and L=400, while the tile size is 16×16 and the kernel size is 3×3. In each synthesized image, 1600 bits are hidden inside some selected tiles. The selection is controlled by a secret key. Results show that the stego textures preserve a good visual appearance.



(a) (b) (c) (d)

(e) (f) (g) (h)

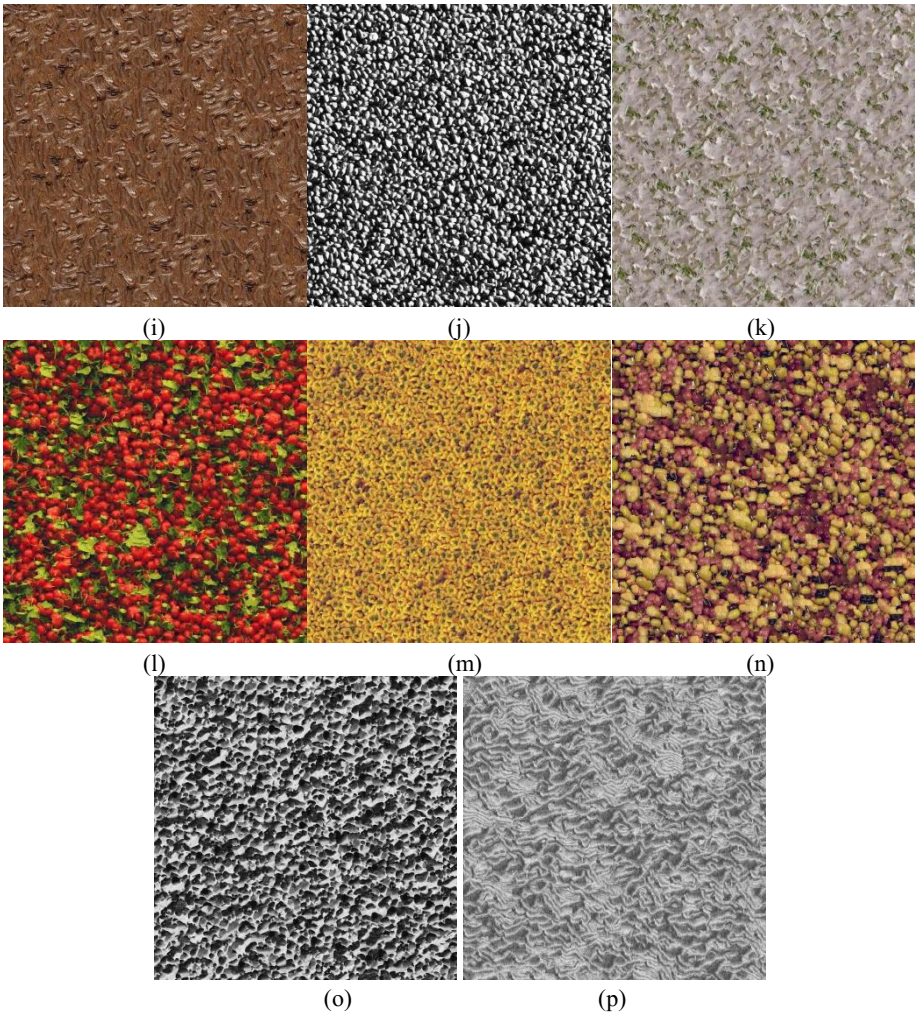


Fig. 5 Source patches and the stego textures. (a)~(h) are the source patches. (i)~(p) are the synthesized textures containing secret messages.

The proposed method is robust to JPEG compression. A group of results are shown in Fig. 6. We use the patterns listed in Fig. 5(a)~(h) to construct stego images. Size of the stego images is 653×653, in which the tile size is 16×16, the kernel size is 3×3, and L=400. Different values of M, from 2 to 16, are used to construct stegos, which are then compressed by JPEG using different quality factors from 10 to 90. After extracting hidden bits from the compressed stegos, the average error rates of the extracted bits in these images are calculated. Results in Fig. 6 show that the average error rate approaches 0 when the quality factor increases. Meanwhile, smaller M results in smaller error rate. When M=2, error rate is equal to zero even if the quality factor of compression is as poor as 10.

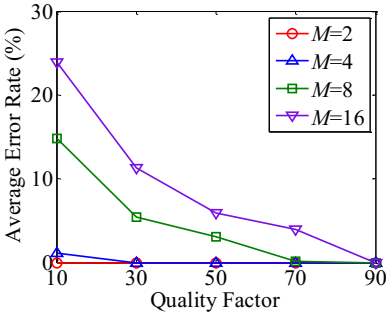


Fig. 6 Error rate of data extraction corresponding to quality factors of JPEG compression

We also use 80 source patches arbitrarily chosen from the database "Brodatz Textures" [5]. These images are rescaled to patterns with the size of 128×128. Parameters are selected to construct stego images with the embedding rate 0.5 bits per tile (non-overlapping) using both the proposed method and the method in [4]. We compress the stego images using different quality factors. Hidden data are extracted from the compressed images and the average error rates are calculated. The results are listed in Table I, indicating that the proposed steganography method has a good capability of countering JPEG compression. As the method in [4] was proposed for steganography in lossless transmission, it is therefore not good at countering JPEG compression.

Table I Average error rate of data extraction after compressing the stego images

Quality Factor	90	70	50
Proposed	0	5.6%	7.7%
[4]	14.5%	24.0%	26.3%

As the stego images are constructed by texture synthesis, sometimes it is somewhat weird to transmit a texture over internet. In real applications, the stegos can be used as the backgrounds of some images. Two examples are shown in Fig. 7. Background of Fig. 7(a) is the synthesized grass containing secret message, adding a football as the foreground. The secret message is hidden in the content outside the blue square. In Fig. 7(b), the synthesized stego texture is used as the background of a pop star, in which secret messages can be extracted from the regions squared by red rectangles.

(a) (b)

Fig. 7 Applications of texture synthesis based steganography. Backgrounds of both (a) and (b) are the synthesized stego textures.

4 Conclusions

This paper proposes a novel steganography method based on texture synthesis. Secret message is represented by different types of tiles captured from a source pattern with small size. Controlled by a secret key, tiles containing secret bits are painted onto the assigned positions in a blank canvas. With image quilting, this painted canvas is then filled with proper tiles to construct a texture image with good visual appearance. After capturing the tiles containing secret bits from the stego, hidden message can be extracted due to the complexities of the tiles. Different from traditional steganography methods, the proposed method provides an approach robustness and large payloads. Since complexities of the tiles containing secret bits slightly change after compression, data extraction is robust to JPEG compression. As a new camouflage way, stego images by texture synthesis can be used in many applications.

Acknowledgement

This work was supported by Natural Science Foundation of China (Grant U1536108, Grant 61572308, Grant 61572452 and Grant 61472235), 2015 Shanghai University Filmology Summit Research Grant, Shanghai Nature Science Foundation (Grant 14ZR1415900), and the Open Project Program of the State Key Lab of CAD&CG (A1502). Corresponding author: Zhenxing Qian, E-mail: zxqian@shu.edu.cn.

References

1. J. Fridrich, Steganography in digital media: principles, algorithms and applications. Cambridge, U.K.: Cambridge Univ. Press, 2009.
2. H. Otori and S. Kuriyama, Data-embeddable texture synthesis, in Proc. of the 8th International Symposium on Smart Graphics, Kyoto, Japan, 2007, pp. 146-157.
3. H. Otori and S. Kuriyama, Texture synthesis for mobile data communications, IEEE Comput. Graph. Appl., vol. 29, no. 6, pp. 74-81, 2009.
4. K-C Wu and C-M Wang, Steganography using reversible texture synthesis, IEEE Trans. Image Process., 24(1): 130-139, 2015.
5. A. A. Efros and W. T. Freeman, Image quilting for texture synthesis and transfer, Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn., pp. 341-346, 2001.
6. Brodatz Textures, Texture Image Database [Online]. Available: http://www.ux.uis.no/tranden/brodatz.html. 1997.
7. W. Zhang, X. Zhang, and S. Wang, Near-Optimal Codes for Information Embedding in Gray-Scale Signals, IEEE Trans. Information Theory, 56(3), pp. 1262–1270, 2010.
8. J. Fridrich and J. Kodovsky, Rich Models for Steganalysis of Digital Images, IEEE Trans. Information Forensics and Security, 7(3), pp. 868-882, 2012.
9. Y.-Q. Shi, C. Chen, and W. Chen, A Markov Process Based Approach to Effective Attacking JPEG Steganography, Proceedings of the 8th International Workshop of Lecture Notes in Computer Science, LNCS, 4437, pp. 249-264, 2007.