Bit-Plane Decomposition Steganography Using Wavelet Compressed Video

Tomonori Furuta, Hideki Noda, Michiharu Niimi, Eiji Kawaguchi
Kyushu Institute of Technology, Dept. of Electrical, Electronic and Computer Engineering,
1-1 Sensui-cho, Tobata-ku, Kitakyushu, 804-8550 Japan

Abstract

This paper presents a steganography method using lossy compressed video which provides a natural way to send a large amount of secret data. The proposed method is based on 3-D set partitioning in hierarchical trees (SPIHT) algorithm for video compression and bit-plane complexity segmentation (BPCS) steganography. In 3-D SPIHT coding, wavelet coefficients in 3-D discrete wavelet transformed video are quantized into a bit-plane structure and therefore BPCS steganography can be applied in the wavelet domain. Embedding rates of around 28% of the compressed video size were achieved for twelve bit representation of wavelet coefficients with no noticeable degradation in video quality. Keywords: steganography, information hiding, information security, lossy compression, wavelet, video

1. Introduction

Steganography is the practice of hiding or camouflaging secret data in an innocent looking dummy container. This container may be a digital still image, audio file, video file, or even a printed image. Once the data has been embedded, it may be transferred across insecure lines or posted in public places. Therefore, the dummy container should seem innocent under most examinations.

In previous steganographic algorithms, bit-plane decomposition was commonly used combined with the simple approach of replacing the binary data in the least significant bit-planes of a dummy image with secret binary data[1]. We previously presented a sophisticated steganography method, called bit-plane complexity segmentation (BPCS) steganography, which makes use of bit-plane decomposition and the characteristics of the human vision system[2],[3]. Noting that human cannot perceive any shape information in a very complicated binary pattern, we can replace noise-like regions in the bit-planes of the dummy image with secret data without deteriorating the image quality. The original BPCS steganography could not be applied to lossy compressed images. However it has been developed recently to be applicable to wavelet-based lossy compressed images including JPEG2000 encoded images[4],[5]. The wavelet-based BPCS steganography enables us to use steganography in a practical scenario where images are compressed before being transmitted.

This paper presents a steganography method using lossy compressed video which provides a natural way to send a large amount of secret data. The proposed method is based on three-dimensional (3-D) set partitioning in hierarchical trees (SPIHT) algorithm for video compression[6] and BPCS steganography. The 3-D SPIHT was proposed for video compression by extending 2-D SPIHT algorithm for image compression[7]. In the 3-D SPIHT, 3-D discrete wavelet transform (DWT) is applied to a video data. The wavelet coefficients in 3-D subbands are then bit-plane-encoded into a bit-stream by 3-D SPIHT encoder. Since bit-planes for wavelet coefficients can be constructed, BPCS steganography can be applied to embed data in the wavelet domain.

The rest of this paper is organized as follows. In Section 2, necessary background on BPCS steganography is given, followed by information on 3-D SPIHT compression in Section 3. In Section 4, BPCS steganography combined with 3-D SPIHT compression scheme is presented. The paper continues with experiments done with an implementation of the 3-D SPIHT-BPCS steganography in Section 5. Conclusions are addressed in Section 6.

2. BPCS Steganography

BPCS steganography uses bit-plane decomposition. When an image is decomposed into bit-planes, we can get binary image for each bit-plane. Fig. 1 shows 5-th and 8-th bit-planes of 8-bpp (bit per pixel) gray image "Girl". It can be seen that the 8-th plane consists of almost all noise-like regions, while the 5-th plane consists of noise-like regions and informative (not noise-like) regions. These random-seeming regions in each bit-plane can be replaced with secret data, which is ideally also noise-like. Because it is difficult for the human eye to distinguish differences between the two noise-like areas, we can disguise the changes to the image.

In BPCS steganography, a complexity measure is introduced to decide whether a binary image is noise-like or not. The complexity measure currently used is defined based on the length of non-edge border between zero and one. The complexity $\alpha$ for an $m \times m$ size binary image is defined as

$$\alpha = \frac{k}{2m(m-1)}, \quad 0 \leq \alpha \leq 1,$$

where $k$ is the total length of the zero-one border in the image and $2m(m-1)$ is the maximum possible border length obtained from an $m \times m$ checkerboard pattern.

1For example, an $n$-bit image can be decomposed into a set of $n$ binary images by bit-slicing operations.
In Fig. 2, white represents a one and black a zero. Both squares, or patches, have the same number of ones and zeros, but very different complexities. This shows that one contains much more visual information than the other. The complex patch (a) has very little visually informative information, therefore it can be replaced with secret data and have a very small effect on the image’s quality. However, if the more visually informative patch (b) was replaced, it would cause noise-like distortion of the definite edges and shapes.

![Figure 1. An example of bit-plane decomposition.](image)

**Figure 2. Noise-like patch (a) and informative patch (b): (a) complexity 68/112, (b) complexity 29/112.**

A typical procedure for data hiding in BPCS steganography is summarized as follows.

1. Segment each bit-plane of a dummy image into small size, for example $8 \times 8$, blocks. Then classify these blocks into informative or noise-like blocks using a threshold of the complexity denoted by $\alpha_0$. A typical value of $\alpha_0$ is 0.3.

2. Segment a secret file into a series of blocks each containing 8 bytes of data. These blocks (which we call secret blocks) are regarded as $8 \times 8$ binary images.

3. If a secret block is less complex than the threshold $\alpha_0$, conjugate it to make it more complex. Here the process called conjugation, which guarantees that any secret data can be embedded, is the exclusive OR operation with a checkerboard pattern. The relation $\alpha^* = 1 - \alpha$ holds true[2], where $\alpha$ and $\alpha^*$ are the complexity of a given image and that of the conjugated image, respectively.

4. Replace each noise-like block in the bit-planes with a block of secret data. If the block is conjugated, then record this fact in a conjugation map.

5. Also embed the conjugation map in the same way as the secret blocks.

The decoding procedure to extract the embedded secret data is just the reverse of the embedding procedure. In the decoding process, the complexity threshold $\alpha_0$ and the amount of secret data need to be known. The amount of secret data can be embedded into a specific place in the dummy file.

In this paper, bit-planes (in fact bit-cubes) for wavelet coefficients in 3-D wavelet transformed video are used to embed data by the BPCS method. Using a complexity measure for 3-D binary images[2], we can embed secret data directly into 3-D binary images. However we took a different approach of using 2-D binary images which constitute a 3-D binary image; secret data is embedded into 2-D binary images based on the complexity measure for 2-D images. This is because the 2-D approach could provide a slightly better performance on embedding capacity than the 3-D approach.

The complexity $\alpha$ for an $m \times m \times m$ size 3-D binary image is defined as $\alpha = \frac{k}{3m^2(m-1)}$, $0 \leq \alpha \leq 1$. Here $k$ is the total area of the zero-one border in the 3-D image and $3m^2(m-1)$ is the maximum possible border area obtained from an $m \times m \times m$ 3-D checkerboard pattern.
3. 3-D SPIHT Video Coding

3-D SPIHT for video coding was proposed extended from 2-D SPIHT for image coding. 3-D SPIHT has the following characteristics: (1) partial ordering by magnitude of 3-D wavelet transformed video, (2) ordered bit-plane coding, and (3) exploitation of self-similarity across spatio-temporal orientation trees (See Fig. 3). Spatio-temporal orientation trees are groups of 3-D wavelet coefficients organized into trees rooted in the lowest frequency (coarsest scale) subband with offspring in several generations along the same spatio-temporal orientation in the higher frequency subbands. Fig. 3 shows parent-offspring relationships in the case of two-level 3-D wavelet decomposition. In the spatio-temporal orientation trees, each node has no offspring or eight offspring, which form a group of $2 \times 2 \times 2$ adjacent pixels. The trees were introduced to exploit self-similarity and magnitude-localization properties in a 3-D DWT video. Typically in case magnitude of a wavelet coefficient in a node does not exceed a given threshold, it is very possible that none of its descendants will exceed that threshold.

![Figure 3. Spatio-temporal orientation tree for the two-level 3-D DWT.](image)

The successive approximation method used by the 3-D SPIHT algorithm encodes the wavelet coefficients one bit-plane at a time, starting with the most significant bit. In 3-D SPIHT compression, each wavelet coefficient $w$ is expressed as

$$w = T( a_0 + a_1 2^{-1} + \cdots + a_{n-1} 2^{-n+1}) , \quad a_i \in \{0, 1\}$$

(2)

where $T = 2^\lfloor \log_2 w_{\text{max}} \rfloor$ ($w_{\text{max}}$ is the maximum absolute value among all wavelet coefficients in a 3-D DWT video). Since $(a_0 + a_1 2^{-1} + \cdots + a_{n-1} 2^{-n+1})$ is a binary expression, the 3-D DWT video can be considered to have a bit-plane structure. 3-D SPIHT encoding is conducted from higher bitplanes to lower ones. That is to say, encoding starts with more important information, so that decoding can be performed on the most important information first. Therefore, even if decoding is discontinued before the end of the video file, almost optimal decoding results can still be achieved under the reduced amount of information. Thus, this type of compression, which is called progressive compression, is particularly suitable for Internet communication.

4. BPCS Steganography Integrated with 3-D SPIHT Video Compression

In the 3-D SPIHT video compression, the wavelet coefficients of a video are quantized into a bit-plane structure and therefore BPCS steganography can be applied in the wavelet domain. The wavelet coefficients have many image-like properties, and BPCS steganography is ideal for exploiting them. The main properties leveraged for BPCS steganography are:

- Correspondence: Spatial areas in each section of the subbands correspond directly to areas in the original image.
- Complexity: The bit-planes at corresponding significance levels of the wavelet coefficients and the original image are usually proportionally complex.
- Resilience: Changes in the values of the wavelet coefficients do not create disproportionately large changes in the reconstructed image.

The procedure for data embedding and extraction in 3-D SPIHT-BPCS steganography is shown in Fig. 4. The entire process to embed data in 3-D SPIHT-BPCS steganography follows the solid line arrows in Fig. 4. After 3-D DWT is applied to an original video, 3-D SPIHT encoder is applied to the wavelet coefficients and a bit-stream (compressed video file) is produced. Then the bit-stream is decoded by 3-D SPIHT decoder and quantized wavelet coefficients are derived. Using these quantized wavelet coefficients, bit-planes for the wavelet coefficients can be constructed and used to embed secret data by BPCS steganography (See the upper box of the right part in Fig. 4). The quantized wavelet coefficients modified by embedding are then subjected to 3-D SPIHT encoding again to produce a secret-data-embedded bit-stream. The symbol (E) in Fig. 4 depicts that secret data is embedded. Data embedding in an already compressed video file is also possible. In this case, the process starts with a compressed video file, i.e., a bit-stream from the bottom of the middle part in Fig. 4 and follows the same process as the aforementioned one.

The data extraction procedure follows the dashed arrows in Fig. 4. 3-D SPIHT decoding of secret-data-embedded bit-stream produces secret-data-embedded quantized wavelet coefficients. Extraction of secret data is carried out by the BPCS method using the quantized wavelet coefficients. We assume that the data extraction starts after the entire file of the bit-stream has been received.

5. Experimental Results

The 3-D SPIHT-BPCS steganography algorithm was implemented and tested on two standard videos: "Claire" and "Image 307x124 to 403x124". In principle the two steps of 3-D SPIHT encoding and 3-D SPIHT decoding are unnecessary to obtain the quantized wavelet coefficients. However, the two steps are performed so that the bit-stream may be truncated to meet pre-embedding compression rate requirements.
Figure 4. A flowchart of data embedding and extraction in 3-D SPIHT-BPCS steganography.

"Diskus". They consist of 32 frames, each of which is 8-bpp gray image and 256 × 256 pixels in size. A four-level 3-D wavelet transform with the Daubechies 9/7 filter was applied to videos. The number of bit-planes in the 3-D SPIHT compression was set to 11 and 12. Here 4 × 4 patch size was used as an embedding unit and random binary data was used as secret data. The complexity threshold $\alpha_0$ for embedding was set to 0.3.

Table 1 shows results of embedding experiments where degradation in video quality has not been perceived after embedding, and compression results without embedding for reference. The PSNR value in the table is the mean for total 32 frames. The least significant bit-plane and the two least significant bit-planes were used to embed data for the number of bit-planes 11 and 12, respectively. The average embedding rate (embedded data size)/(compressed video file size) for two videos was around 18% for 11 bit-planes, and 28% for 12 bit-planes. Fig. 5 shows frames for "Claire" (upper row) and "Diskus" (lower row) whose PSNRs after embedding are the lowest among 32 frames. In Fig. 5, (a) and (d) are original frames, (b) and (e) are compressed frames without embedding, and (c) and (f) are embedded and compressed frames. The PSNRs for (c) and (f) are 43.3dB and 39.4dB, respectively.

6. Conclusions

This paper presented a large capacity steganography method applicable to compressed video, which is invented based on BPCS steganography and 3-D SPIHT video compression. The proposed 3-D SPIHT-BPCS steganography achieved embedding rates of around 28% of the compressed video size for twelve bit representation of wavelet coefficients with no noticeable degradation in video quality.

We should note that the 3-D SPIHT-BPCS scheme is not robust with respect to lossy compression in the sense that lossy compression of an already embedded and compressed dummy video can still easily destroy the embedded information. Works are under progress to realize another video-based steganography applicable to Motion-JPEG2000 compressed video.

Acknowledgement This work was partly supported by the International Communications Foundation.

7. References

Table 1. Results of embedding experiments

<table>
<thead>
<tr>
<th>video</th>
<th># bit-planes</th>
<th># bit-planes used for embedding</th>
<th>embedded data size (bytes)</th>
<th>compressed file size (bytes)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claire</td>
<td>11</td>
<td>-</td>
<td>118175</td>
<td>123678</td>
<td>47.1</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>2</td>
<td>249508</td>
<td>264709</td>
<td>49.7</td>
</tr>
<tr>
<td>Diskus</td>
<td>11</td>
<td>-</td>
<td>316656</td>
<td>326881</td>
<td>44.2</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>2</td>
<td>540452</td>
<td>567980</td>
<td>48.8</td>
</tr>
</tbody>
</table>

Figure 5. Examples of experimental results: (a),(d) original frames, (b),(e) 3-D SPIHT compressed frames, (c),(f) embedded and 3-D SPIHT compressed frames.