Robust steganographic method utilizing properties of MJPEG compression standard

Jakub Oravec
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics
Technical University Košice
Košice, Slovakia
jakub.oravec@tuke.sk

Gabriel Bugár, Ján Turán
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics
Technical University Košice
Košice, Slovakia
gabriel.bugar@tuke.sk; jan.turan@tuke.sk

***Abstract*—This article presents design of steganographic method, which uses video container as cover data. Video track was recorded by webcam and was further encoded by compression standard MJPEG. Proposed method also takes in account effects of lossy compression. The embedding process is realized by switching places of transform coefficients, which are computed by Discrete Cosine Transform. The article contains possibilities, used techniques, advantages and drawbacks of chosen solution. The results are presented at the end of the article.**

Keywords—steganography, robustness, MJPEG, Discrete Cosine Transform.

I. INTRODUCTION

In these days, when various data are transferred through networks like Internet, the question of data security becomes more and more important. There are two major ways for getting certain level of security: cryptographic methods and steganographic methods. Cryptographic methods are used for changing data in a way that other users are not able to recover original data without knowing secret password, called key. On the other hand, steganographic methods use different pattern – they use other data, called cover data, to hide secret data. Cover data, which were modified by secret data, are called simply stego data, or modified data [1]. In this article, we will propose design of steganographic method, which uses video container as cover data. More exactly, secret data will be hidden in video frames, which were encoded by using compression standard MJPEG (Motion JPEG). This standard is nowadays used for compression of video from IP cameras, or webcams, which was our case. This article also tries to describe some of techniques, which could be used for getting higher level of robustness against lossy compression.

II. TERMINOLOGY

A. Steganographic terms

To avoid confusion about meaning of used terms, we shall briefly describe some of used terms, techniques and principles. Embedding algorithm is algorithm, which creates stego data by hiding (embedding) secret data in cover data. Extraction algorithm is used for inverse operation, which is getting the secret data out of stego data. Steganographic algorithms tend to use public channel, because they rely on fact that users, or machines are not able to trace hidden data. The process of tracing hidden data, by human user, or by machines is called stegoanalysis. Any user, who is not either sending, or receiving stego data, and tries to modify stego data, is marked as attacker, and this action is considered to be an attack. Robustness is feature of steganographic algorithms, which denotes their resistance from attacks, for example from lossy image compression [1]. Other feature is called capacity and it contains information about size of data, which can be hidden in chosen cover data. With higher capacity, the robustness is smaller, and vice versa. Higher robustness is used in digital watermarking systems. Pure steganographic systems desire higher capacity for secret data. Proposed steganographic method tries to achieve compromise between these two properties.

An easiest kind of steganographic system is shown on Fig. 1. Blocks marked by dashed line are optional.

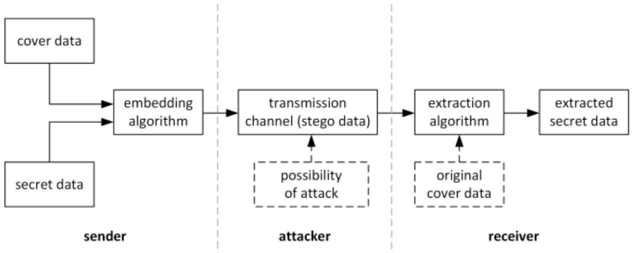


Fig. 1. Example of steganographic system.

B. Digital video terms

Video file, which we watch is created by merging various modalities, such as images, sound, or text used in subtitles. These modalities have to be stored somehow – they use data structure called video container. Frames of video are stored together as a video track. Audio track is a set of audio samples. One video container can have multiple video and audio tracks. Possible structure of video container is shown on Fig. 2.

The rate at which are the image frames changing is called frame rate. Unit of frame rate can be either Hertz (abbreviation Hz), or frame per second (fps). Frames are described by their resolution and color depth. Resolution of video frame consists of the frame width, and its height. These values are mostly denoted in pixels. Color depth of video frame determines number of used color shades. Shade of certain pixel in image is determined by used color model and the value of intensity function.

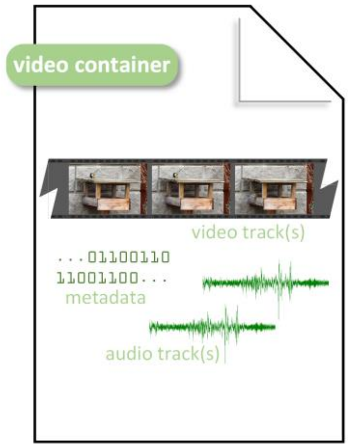


Fig. 2 Contains of video container

The contents of video, or audio tracks also depend on used compression standard. Compression standards for video tracks can be divided by the way which they use to process and encode input data [2]. First group of compression standards use image compression standards on each frame from video track. Second group tries to encode some of frames by using similarities with other frames. Video compression standard MJPEG, which was chosen for our application uses compression standard JPEG (Joint Photographic Experts Group) for encoding each frame of video track.

III. Proposed method

In the design phase, we encountered several problems. We tried to solve these problems by using some of ideas, or techniques which are presented below.

A. *Used video container and compression standard*

We decided to choose AVI (Audio Video Interleave) video container. Despite its age (it was introduced in 1992), this container is still used. AVI video container is well documented [3] and has various properties, which could be used for steganography purposes. For example, AVI container can have multiple metadata fields, which can contain secret data.

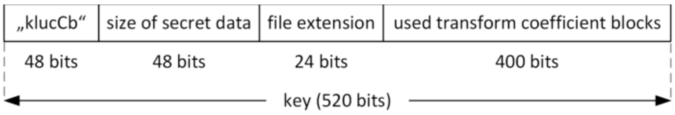
As we have mentioned above, we chose MJPEG as our compression standard. MJPEG compresses and encodes each video frame as still image by using JPEG. In other words, MJPEG does not use inter frame encoding techniques, such as prediction. This feature makes the capacity of cover data higher, as some other compression standards use inter frame encoding. When compression standard uses this type of encoding, steganographic algorithm is not able to use all video frames for secret data embedding. Thus in our solution, every frame is used.

One of major drawbacks of MJPEG is that when block of video frame has certain small amount of energy, it is encoded as block of zeros. In this case steganographic algorithm needs to determine which are these blocks and do not use them while it is embedding secret data into cover data.

B. *Choice of suitable transform coefficient blocks*

As we have mentioned above, some of video frame blocks are not encoded due to their small energy. Because of this property, these blocks could not be used for embedding secret data. In proposed method, the suitability of each block is determined by amplitude of DC transform coefficient. This coefficient is used because Discrete Cosine Transform (DCT) tends to accumulate biggest amount of energy into it [2, 4].

Information about chosen pixels needs to be provided also for extraction of secret data. This is utilized by usage of “key”. The term key is written in quotation marks, because it does not have the same function as it has in cryptography. For our method key is created by several fields, which are illustrated on Fig. 3. First of these fields, “klucCb” is a marker for valid key. Second field determines the size of stored secret data. This field is important, because extraction algorithm needs to determine, if secret data was extracted successfully. Third field contains the file extension of secret data file. Last, fourth field carries information about used transform coefficient blocks.



„klucCb“	size of secret data	file extension	used transform coefficient blocks
48 bits	48 bits	24 bits	400 bits

key (520 bits)

Fig. 3. Used key structure

C. *Y’CbCr color space and chroma subsampling*

Image processing applications use different color spaces for processing color images [1, 5]. These color spaces consist of several planes, which are used for recording the value of each pixel’s intensity function. Because our proposed method uses compression standard MJPEG, we decided to use the same color space, which is used in MJPEG. It is color space Y’CbCr. This color space consists of three planes: one luminance plane Y’ and two chroma planes Cb and Cr. The luminance plane carries information about brightness (intensity) of scene and the chroma planes, known also as chrominance planes, describe the color shade of every pixel in scene. Example of image and its representation in Y’CbCr color space is shown in Fig. 4.

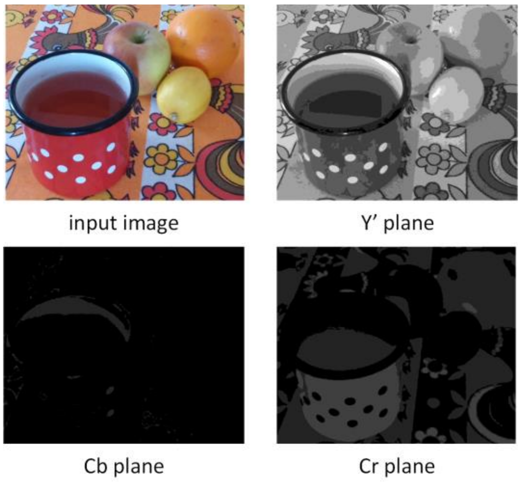

input image
Y’ plane
Cb plane
Cr plane

Fig. 4. Image decomposed in Y’CbCr color space.

Relations between color spaces RGB and Y’CbCr are denoted as systems of equations (1) and (2) respectively.

$$Y'(m) = 0.222 \cdot R(m) + 0.7067 \cdot G(m) + 0.0713 \cdot B(m)$$

$$Cb(m) = \frac{B(m) - Y\prime(m)}{1.8574}$$

(1)

$$Cr(m) = \frac{R(m) - Y'(m)}{1.556}$$

$$R(n) = Y'(n) + 1.556 \cdot Cr(n)$$
$$B(n) = Y'(n) + 1.8574 \cdot Cb(n) \qquad (2)$$
$$G(n) = \frac{Y\prime(n) - 0.0713 \cdot B(n) - 0.222 \cdot R(n)}{0.7067}$$

where *m*, or *n* denote value of intensity functon in color space RGB, or Y’CbCr, respectively.

Another operation, which is used by MJPEG (thus also JPEG), is known as chroma subsampling. Reference [6] provides information, that human visual system is less sensible of chroma distortion, than luminance distortion. Because of this fact, the chroma planes could be subsampled. The process of subsampling can be characterized as degrading chroma information by removing some of values from chroma plane. Subsampling is not a reversible operation, thus after subsampling a chroma plane we are not able to get the original chroma values for reconstructed picture. There are various techniques for subsampling and some of them are described by a factor known as subsampling ratio [1]. Effects of various subsampling ratios are illustrated in Fig. 5. MJPEG uses ratio 4:2:0 for chroma planes Cb and Cr.

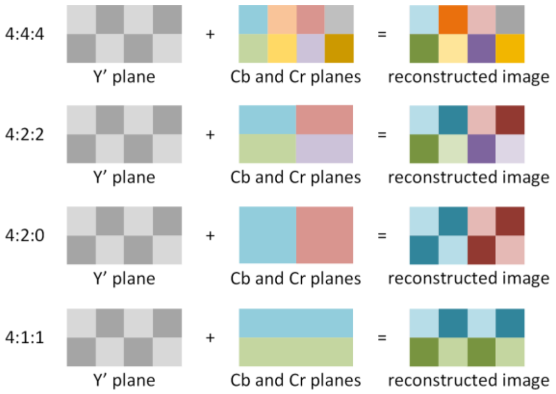


Fig. 5. Demonstration of results of chroma subsampling with various ratios.

D. *Choice of steganographic algorithm*

For achieving certain level of robustness for our steganographic method, we chose steganographic algorithm DCT SWAP [6-8]. This algorithm changes – swaps position of selected transform coefficients in determined manner, for example value of transform coefficient with position (3; 3) should be higher than value of coefficient with coordinates (2; 2). Solutions like this one are known as blind steganographic methods due to fact they do not need to use the original cover data in the extraction algorithm [1, 6]. However, our application uses original cover data for purposes of maximizing the level of robustness against lossy compression.

Changing positions of various transform coefficients can cause distortion, so for our application we decided to switch places of one pair of transform coefficients. Idea of DCT SWAP algorithm and effects on modified image are shown on Fig. 6.

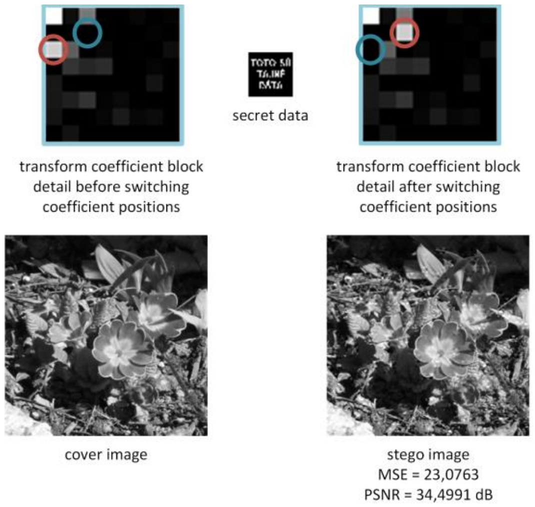

TOTO SÚ TAJNÉ DÁTA
secret data
transform coefficient block detail before switching coefficient positions
transform coefficient block detail after switching coefficient positions
cover image
stego image
MSE = 23,0763
PSNR = 34,4991 dB

Fig. 6. Idea behind and effects of DCT SWAP algorithm.

E. *Embedding algorithm*

Before the embedding process itself starts, some operations are carried out. Firstly, secret data, which is going to be hidden, is loaded as a file, thus as set of characters. Each character is then transformed to binary code using ASCII (American Standard Code for Information Interchange) table. Due to this operation, embedding algorithm will perform only two operations – first one if the value of current bit will be zero, and other one if the value will be one.

Second operation is creation of a “key”. Contents of this key and their applications are described above. Key is used in the extraction algorithm for specification of secret data length. In other words the key determines the frames, which are used for data hiding.

Embedding itself starts by loading cover data, which are in our application video frames. Each frame is converted from color space RGB to color space Y’CbCr. Chroma planes are further subsampled by ratio 4:2:0. Each plane of color space Y’CbCr is then divided into non-overlapping blocks of 8x8 pixels. Two dimensional Discrete Cosine Transform (3) of these blocks is carried out for acquiring energy decomposition in transform domain. Blocks in transform domain are quantized by dividing the values in blocks by values of quantization matrix and rounding them to nearest integer. These operations are similar to operations, which will happen, if image would been encoded with usage of JPEG compression standard [9].

$$F(u,v) = \frac{2 \cdot C(x) \cdot C(y)}{\sqrt{M \cdot N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot cos\left[\frac{\pi u(2x+1)}{2 \cdot M}\right] \cdot cos\left[\frac{\pi v(2y+1)}{2 \cdot N}\right] \qquad (3)$$

$$f(x,y) = \frac{2 \cdot C(u) \cdot C(v)}{\sqrt{M \cdot N}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \cdot cos\left[\frac{\pi u(2x+1)}{2 \cdot M}\right] \cdot cos\left[\frac{\pi v(2y+1)}{2 \cdot N}\right] \qquad (4)$$

$$\text{where } C(k) = \begin{cases} \frac{1}{\sqrt{2}} \; if\; k = 0 \\ 1 \; if \; k \neq 0 \end{cases},$$

F(u, v) is value of transform coefficient with position *(u, v)*, *MxN* is resolution of transformed block and *f(x, y)* is value of intensity function of pixel with position *(x, y)*.

Embedding of secret data itself is realized by switching places of two quantized values in block of 8x8 subsampled values of chroma plane Cb. We chose to change positions of values with positions (2; 3) and (3; 1). Switching of quantized values is made only if the values do not meet selected condition. We used the condition that value with position (2; 3) has to be higher than value with position (3; 1). For reaching certain level of robustness, this condition must be fulfilled also after lossy compression.

After these steps, inverse operations are performed. Quantized values are dequantized by multiplying them with values from quantization matrix. Two dimensional Inverse Discrete Cosine Transform (4) is carried out and original video frame is reconstructed from color space Y'CbCr. At this point the stego data is saved to the hard disk.

F. Extraction algorithm

After user provides key and its information is loaded, original cover data is being used for embedding two different sequences with the same length as the secret data has. The first one consists only of zeros and the second one of ones. After embedding these sequences, the blocks of transform coefficients of Cb plane are taken, and are matched with blocks from stego data. The level of resemblance is calculated by using Euclidean distance (5).

$$d = \sqrt{(a - b)^2} \tag{5}$$

where *a* and *b* are values of compared transform coefficients.

Value of secret data bit is determined by the means of smaller Euclidean distance. This operation is repeated until last bit is reached. If last bit could not be reached, extraction algorithm ends and gives user a warning about corrupted stego data.

The extraction process may seem too complicated for a blind steganographic algorithm as DCT SWAP, but the usage of Euclidean distance and original cover data is made for purposes of achieving better results against certain attacks – mainly lossy compression.

IV. Properties of proposed method

Used technologies and the way we designed our method result in several properties of this method. Firstly, the blocks of transform coefficients, which will contain secret data are chosen before the cover data in form of video are recorded. Changes of recorded scene either during the recording, or before it could lead to the situation, when block of transform coefficients is not suitable for secret data embedding anymore. After the embedding, visible artifacts can be created. This situation is shown in Fig. 7.

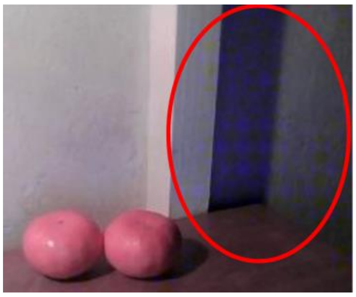


Fig. 7. Detail of visible artifacts as result of recorded scene change.

Second problem is caused by the form of secret data – it is a file, thus it contains all forms of data fields. For example metadata or headers. Data fields such as last mentioned are very sensitive to changing their values. Because of this fact, some extracted secret files could become corrupted. This happens especially after lossy compression with smaller values of quality factor.

V. Experimental results

We made several experiments to demonstrate effects and properties of our proposed steganographic method. We have used the same video container in all of experiments. Video track was recorded by built-in webcam. Secret data file was in form of BMP image. A lossy compression of stego data was carried out, with various quality factors (Q = 100; 90; 80). Resulting extracted images are shown on Fig. 8. We have also computed parameters, which are used for determining extracted secret data quality [10]. These are Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). Equations used for their calculation can be found in for example [1].

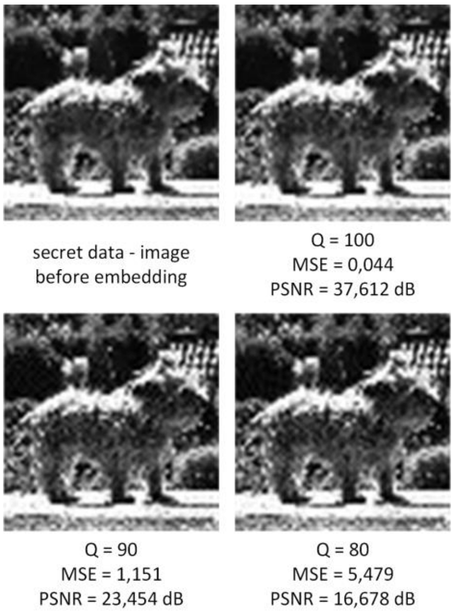


Fig. 8. Extracted images after lossy compression

Proposed method proved to be robust against lossy compression to quality factor as low as Q = 75. If this value is less, extraction of secret file is still successful, but some of data fields tend to be damaged. If damaged field is particularly important, the extracted secret file can become corrupted. This happened in our experiment with Q = 70, when the image data was extracted without any error, but extracted image had different resolution – contents of this data field were not correct.

VI. Conclusion

In this article we have proposed a steganographic algorithm, which uses cover data in form of video container. Secret data, which are being embedded are in form of various files. We have used compression standard MJPEG, which allowed certain level of capacity, which is important for steganographic algorithms. We have discussed about properties of such solution and provided some of experimental results.

Acknowledgment

This work was supported by following research grants: KEGA 006TUKE-4/2014, VEGA: 1/0075/15 and APVV-0025-12.

References

[1] D. Levický, „Multimédia a ochrana ich obsahu“ elfa, Košice, 2012. ISBN: 978-80-8086-199-5, 249 pp.

[2] J. Mihalík, “Kódovanie obrazu vo videosekvenciách” elfa, Košice, 2001. ISBN: 80-89061-47-8, 244 pp.

[3] A. Noe, “AVI File Format”. [online], 2006. [cit. 4. 10. 2015]. Available on: http://www.alexander-noe.com/video/documentation/avi.pdf.

[4] N. Ahmed, T. Natarajan, K. R. Rao, “Discrete Cosine Transform” in IEEE Transactions on Computers, 1974, pp. 90-93.

[5] S. Katzenbeisser, F. A. Petitcolas, “Information Hiding – techniques for steganography and digital watermarking” Artech House, Boston, 2000. ISBN: 978-15-8053-035-4, 220 pp.

[6] F. A. Petitcolas, R. J. Anderson, M. G. Kuhn, “Information Hiding – A Survey” in Proceedings of IEEE, special issue on protection of multimedia content, 1999, pp. 1062-1078.

[7] K. Ramanjaneyulu, P. Pandarinath, B. Rakesh Reddy, “Robust and Oblivious Watermarking based on Swapping of DCT Coefficient” in International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2013, pp. 445-452.

[8] A. Parnami, A. Gupta, G.Parnami, “A Robust DCT based Digital Image Watermarking using Random Mid-band Coefficient Exchange Scheme for Gray Scale Images” in International Journal of Computer Applications, No. 8, Volume 100, 2014, pp. 6-10.

[9] G. K. Wallace, “The JPEG Still Picture compression standard” in IEEE Transactions on Consumer Electronics, February 1992, pp. 1-17.

[10] A. Samcovic, J. Turán, “Attacks on Digital Wavelet Image Watermarks” in Journal of Electrical Engineering, Vol. 59, No. 3, 2008, pp. 131-138.

Biographies

Jakub Oravec (Ing.) received Ing. (MSc.) degree in 2015 at Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics of Technical University of Košice. Since September 2015 he has been at University of Technology, Košice as PhD. student. His research interests include steganography, digital watermarking and digital image processing.

Gabriel Bugár (Ing., PhD.) received Ing. (MSc.) degree in Electronics and Communication Techniques from the Technical University, Košice, in 2007. He received PhD. degree from Technical University, Košice, Slovakia, in 2010. Since September 2010, he has been at the Technical University, Košice as Assistant Professor. His general research interests include image processing, steganography and steganalysis.

Ján Turán (Dr. h. c., prof., RNDr., Ing., DrSc.) received Ing. (MSc.) degree in physical engineering with honours from the Czech Technical University, Prague, Czech Republic, in 1974, and RNDr. (MSc.) degree in experimental physics with honours from Charles University, Prague, Czech Republic, in 1980. He received a CSc. (PhD.) and DrSc. degrees in radioelectronics from Technical University, Košice, Slovakia, in 1983, and 1992, respectively. Since March 1979, he has been at the Technical University, Košice as Professor for electronics and information technology. His research interests include digital signal processing and fiber optics, communication and sensing.