Applied Computer Systems doi: 10.1515/acss-2016-0007

2016/19

Comparison of Video Steganography Methods for Watermark Embedding

David Griberman[1], Pavel Rusakov[2]

[1, 2] *Department of Applied Computer Science, Riga Technical University, Latvia*

***Abstract* – The paper focuses on the comparison of video steganography methods for the purpose of digital watermarking in the context of copyright protection. Four embedding methods that use Discrete Cosine and Discrete Wavelet Transforms have been researched and compared based on their embedding efficiency and fidelity. A video steganography program has been developed in the Java programming language with all of the researched methods implemented for experiments. The experiments used 3 video containers with different amounts of movement. The impact of the movement has been addressed in the paper as well as the ways of potential improvement of embedding efficiency using adaptive embedding based on the movement amount. Results of the research have been verified using a survey with 17 participants.**

***Keywords* – Copyright protection, data security, digital watermarking, steganography, video processing.**

I. Introduction

The word “steganography” comes from the Greek language and means “covered writing”. Steganography as a science studies the exchange of information in a way that the fact of the exchange remains unseen [1]. Similarly to cryptography, the goal of steganography is to protect information, but unlike cryptography, where the existence of information is not hidden, steganography aims at concealing it behind a cover – a stegocontainer. Modern digital steganography embeds the message (a sequence of bits) into a container (another sequence of bits), receiving a stegocontainer as a result – a sequence of bits, similar to the original container, but containing the hidden message. Digital pictures, videos, text documents and other digital files can be used as a container. A simplified steganographic process is shown in Fig. 1.

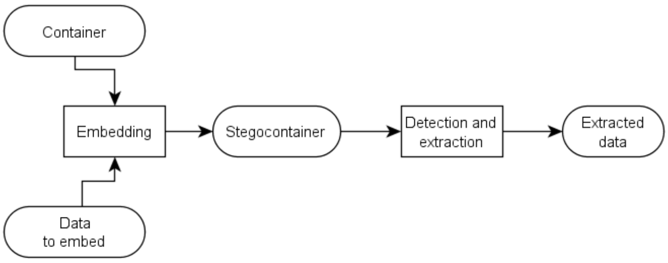


Fig. 1. A simplified steganographic process [2].

Although the classic usage of steganography is hidden communication, modern digital steganography can be used for other purposes, such as copyright protection, content authentication and others [2], [3]. In case of copyright protection, the goal changes – the hidden message is an invisible watermark that identifies the original author or owner of the work. In case of copyright protection, robustness of the hidden message becomes important – it should still be readable after compressing the original work with a lossy compression algorithm as well as after various operations that the end user can perform on it (filtering, cropping etc.), which is usually not as important when the goal is only hidden communication.

The goal of the present paper is to compare various methods of video steganography in the context of copyright protection that can allow for the embedment of an invisible digital watermark into a video file and research possible areas of improvement for these methods.

Section II contains a short description of video steganography and the classification of its methods. Section III describes methods selected for the experiment as well as criteria defined for the selection, while section IV describes the experiment itself. Section V contains the first results of the experiment, followed by a description of possible improvements using an adaptive approach in Section VI. Section VII is devoted to test the robustness of these methods under compression and geometric attacks. Section VIII describes and gives results of a survey conducted to verify the efficiency of the method. Section IX features related studies in the field. The last section provides a summary of the present paper as well as defines possibilities for further research.

II. Video Steganography

Video steganography is the field of digital steganography that is less researched than image steganography, yet it provides a set of benefits [4]:

- the changes are less noticeable for the human visual system as frames are visible only for a short duration of time;
- frames individually can be less clear and focused, small colour changes are not as noticeable compared to photography.

Video steganography methods can be grouped in various ways [5]:

- By compression:
  - methods for compressed video files;
  - methods for uncompressed (raw) video files.
- By embedding domain:
  - spatial domain;

  - transform domain (discrete cosine transform – DCT, discrete wavelet transform – DWT, etc.).
- By embedding method:
  - embedding into still images;
  - embedding based on the video codec;
  - embedding based on the video format.

As most of the videos on the World Wide Web are compressed video files and a raw video file alone can raise suspicion [6], the paper focuses on methods that are used with compressed video files.

III. Embedding Methods

For the purposes of the research, the following criteria have been defined for video steganography methods:

1. The method must support compressed video containers – uncompressed videos are uncommon and may raise suspicion.
2. Graphical watermark embedding support – even with noise added it can still be identified visually. As shown in Fig. 2, grayscale watermarks used in the experiments feature the letters "RTU" and the logo of Riga Technical University.
3. No video file format or codec dependency – otherwise the attacker (or the user) can change them to destroy the embedded watermark.

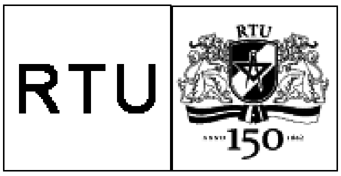


Fig. 2. Watermarks used in the experiments (90 x 90 pixels).

Methods that satisfy the stated criteria should be able to encode a watermark that is still readable after uploading the stegocontainer to video hosting services on the World Wide Web – like Youtube or Facebook, where additional processing is applied to it. As the bandwidth of the internet is growing, online video services are becoming more popular and give additional opportunities to embed data in online sites like Facebook [7].

At first, publicly available video steganography tools were reviewed – MSU StegoVideo [8], OpenPuff [9], RTStegoVideo [10] and Steganosaurus [11]. None of them could successfully embed data in a video container that could still be readable after uploading to a video service.

Steganosaurus and MSU StegoVideo were able to partially embed the data, yet it was lost after recompression. Both tools also have video codec dependencies. RTStegoVideo is meant for streaming and was unable to embed in a video file for upload. OpenPuff embeds the message in redundant data of the video (possibly metadata, based on the speed), but the redundant data are lost after recompression, making it unsuitable for the purposes of the research.

Four video steganography methods from scientific papers were chosen as an alternative (see Table I for short descriptions and original sources) – two of them were originally image steganography methods that the authors adapted for video steganography (a video can be considered a sequence of frames). As none of these methods have publicly available implementations they were implemented by the authors in the form of Java program.

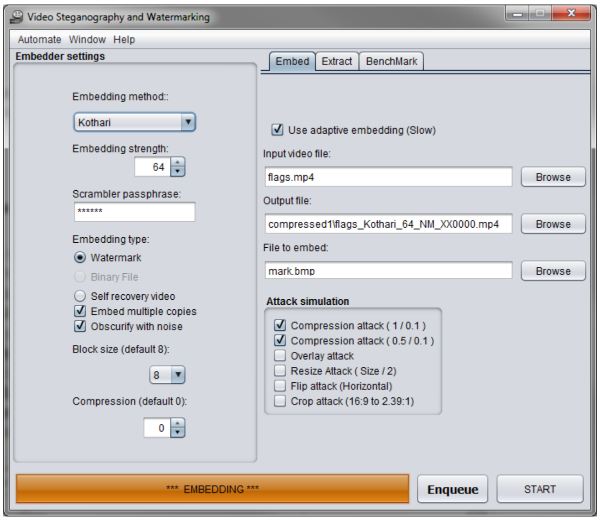
TABLE I

Embedding Methods Selected

Label and Source	Short Description
Kaur [12]	Method is based on embedding data into 8 x 8 DCT coefficients of a frame in the Y channel of the YCbCr colour space. Two coefficients were selected and compared – depending on which one had a larger value, a value of 1 or 0 was decoded. The embedding strength Δ shows the minimum difference between the two coefficients.
Kothari [13]	Method is similar to the Kaur method, but uses different coefficients and operates on the G channel of the RGB colour space.
Dubai [14]	Method is based on embedding data into 8 x 8 DCT block coefficients using paired and unpaired quantization. G channel of the RGB colour space was used for the experiments. Δ shows the precision of quantization.
Haar [15]	The method embeds into LL, HH and HL regions of the DWT transformation. Y channel of the YCbCr colour space was used for the experiments. Δ shows the level of DWT transformation used.

IV. Experiment Description

A video steganography tool that features implementations of the four methods described previously has been developed for the experiment (see Fig. 3.) It is based on the Xuggler [16] library and was written in Java for cross-platform support. Experiments were made by embedding into .mp4 format files with H.264 and MPEG4 codecs (the software supports all formats and codec that are supported by Xuggler, yet some are not suited for embedding).



Embedder settings

Embedding method:: Kothari

Embedding strength: 64

Scrambler passphrase: ******

Embedding type:
● Watermark
○ Binary File
○ Self recovery video
☑ Embed multiple copies
☑ Obscurify with noise

Block size (default 8): 8

Compression (default 0): 0

☑ Use adaptive embedding (Slow)

Input video file: flags.mp4

Output file: compressed1\flags_Kothari_64_NM_XX0000.mp4

File to embed: mark.bmp

Attack simulation
☑ Compression attack (1 / 0.1)
☑ Compression attack (0.5 / 0.1)
□ Overlay attack
□ Resize Attack (Size / 2)
□ Flip attack (Horizontal)
□ Crop attack (16:9 to 2.39:1)

Fig. 3. Main window of the program developed.

Three video containers were used, each with different amount of movement (see Table II for parameters and Fig. 4 for illustrations):

- riga.mp4 – a panorama of the city of Riga, the entire frame is almost still (low movement);
- flowers.mp4 – red flowers slowly moving in the wind (medium movement);
- flags.mp4 – coloured flags moving around in the wind quickly (high movement).

The video format was chosen as a compromise between embedding speed, video quality and the popularity of the file and codec format on the internet.

TABLE II

PARAMETERS OF VIDEO CONTAINERS USED

Resolution	1280x720 pixels	**Length (seconds)**	~ 4
Video codec	H.264	**Frames per second**	25.00
Container format	MP4	**KB per second**	~ 6350

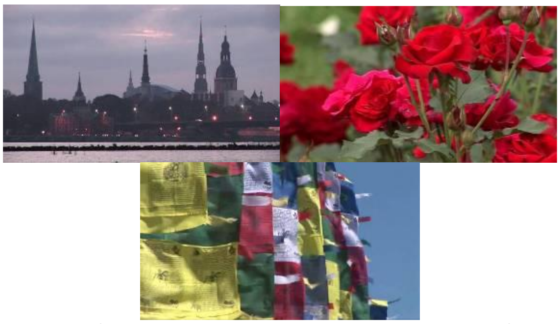


Fig. 4. Frames from the video containers used (riga.mp4, flowers.mp4, flags.mp4).

Before embedding, the watermark was prepared as shown in Fig. 5. At first, the watermark was transformed to a black and white image and then to a sequence of bits. After that it could be duplicated to fill up all of the available space in the container. Remaining space could be filled with random data and the data to be embedded could be shuffled using a password.

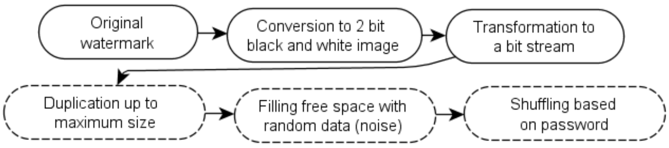

Original watermark
Conversion to 2 bit black and white image
Transformation to a bit stream
Duplication up to maximum size
Filling free space with random data (noise)
Shuffling based on password

Fig. 5. Steps for the preparation of a watermark for embedding (optional steps marked with a dashed line).

V. Experiment Results

At the first stage of the experiment, watermarks were embedded into every frame of the 3 video files of different embedding strengths using H.264 and MPEG4 video codecs. At first, the compression rate was chosen close to the size of the original video file to minimise its impact on the embedding.

Both the embedding efficiency and the peak signal-to-noise ratio (PSNR) were measured. The embedding efficiency was measured as α, i.e. the number of pixels being the same in both the extracted watermark and the original one divided by the total number of pixels. Watermark was extracted for each frame and an average watermark was calculated as a final result. αFINAL shows the value for the final watermark, αAVERAGE – the average value, αMAX and αMIN demonstrate maximum and minimum values of individual watermarks extracted.

The results are summarised in Tables III and IV. They show the best PSNR for containers from which a readable watermark could be extracted (αFINAL value). Over 500 measurements were made in total.

TABLE III

PSNR VALUES ON WHICH A READABLE WATERMARK WAS EXTRACTED (USING H.264 CODEC)

	Kaur	Dubai	Kothari	Haar
flags	38.9	–	41.34	24.13
flowers	40.62	–	42.38	35.03
riga	41.11	39.8	45.4	36.85

TABLE IV

PSNR VALUES ON WHICH A READABLE WATERMARK WAS EXTRACTED (USING MPEG4 CODEC)

	Kaur	Dubai	Kothari	Haar
flags	35.18	27.54	36.31	24.06
flowers	40.46	–	43.33	35.53
riga	40.34	39.05	44.58	31.12

The first results showed that the Kothari method had the highest PSNR value, followed by the Kaur method. Dubai method had difficulties with embedding into moving video containers – the watermark could not be extracted afterwards. Haar method made noticeable distortions in the video files, mostly in the moving parts of the frame.

Although the resulting file sizes were similar, the H.264 codec showed higher PSNR values for all methods compared – only it was used for further experiments.

VI. The Adaptive Approach

Results of the experiments showed that videos that had a lot of movement were more difficult to embed than videos with little or no movement. This could be explained by the compression algorithms used and the fact that embedding was performed in every single frame. Therefore, the authors suggest a new approach – analysing the frame for movement and varying the embedding strength based on it.

In the experiment, a frame and N succeeding frames were compared by their brightness values – if it were over a threshold of Ψ an area of the frame was considered to be in movement. Figure 6 demonstrates sample frames with detected regions coloured in white. Any other movement detecting algorithm can be used for this approach if it allows measuring the amount of movement in the areas of the frame.

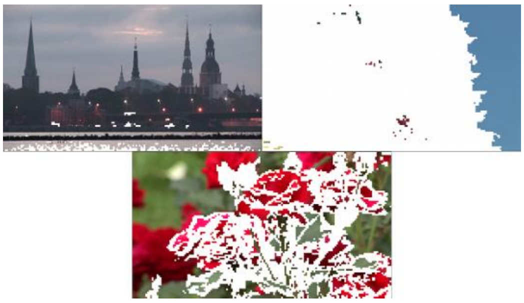


Fig. 6. Example of regions with motion detected (coloured white).

When the areas with movement were identified, adjustments in the embedding process could be made. The authors modified the Kothari and Kaur methods to take into account the amount of movement during embedding as follows:

- If the value of movement for an area of the frame Ψ' is below the threshold, embedding strength is adjusted as in (1);
- If the value is above the threshold do nothing;
- If the value is below the threshold and is neighbouring with a high threshold area (Z) adjust the value of it as in (2).

$$\Psi' < \Psi_1 \rightarrow \Delta_\Psi = \Delta * 0.5 \qquad (1)$$

$$Z(\Psi' \geq \Psi_1) < \Psi_1 \rightarrow \Delta_\Psi = \Delta * 0.75 \qquad (2)$$

The coefficients above were selected based on the contents and type of the videos, the compression algorithms and the embedding methods used. These coefficients need to be adjusted for other situations.

Tables V and VI show the PSNR levels of Kaur and Kothari methods (both adaptive and regular variants) with the embedding strength when they were able to receive αFinal ≥ 0.99 and αMin ≥ 0.9 respectively. As demonstrated by the results, the methods performed better on the flag video for the αFinal value and better on all videos when αMin was measured.

TABLE V

Comparison of Kaur, Kothari and Their Adaptive Methods on the First αFinal ≥ 0.99 Value

	Kaur	Kaur Adaptive	Kothari	Kothari adaptive
flags	38.9	39.24	41.34	41.51
flowers	40.62	40.44	42.38	42.31
riga	41.11	41.07	45.4	46.39

TABLE VI

Comparison of Kaur, Kothari and Their Adaptive Methods on the First αMin≥0.9 Value

	Kaur	Kaur Adaptive	Kothari	Kothari adaptive
flags	34.22	38.04	34.81	38.21
flowers	38.26	38.99	39.75	40.9
riga	40.87	40.87	43.91	45.11

After the experiment, the methods were tested on a combined container that featured all three videos combined into one (with a dissolve effect from black in the beginning and to black in the end). In the authors' opinion, the visual degradation of the video was more noticeable during the dissolve on the monotone black background. A modification in the adaptive algorithms was made to embed less into monotone regions of the image – the impact of which was tested with a survey (results are shown in Section VIII).

VII. Measuring Robustness

Another important parameter for embedding methods is robustness [2] – in the context of digital steganography it measures the degree to which the embedded message is able to resist certain changes in the stegocontainer and still be readable. Without robustness the method is unusable for copyright protection – the attacker can simply remove the embedded watermark by changing the stegocontainer even slightly because the watermark is fragile. Kaur and Kothari methods as well as their adaptive variants were compared by applying different levels of compression to the stegocontainer. Tables VII and VIII show results with compression ratios of 1 (recompressed) and 0.5 (half of the original file size).

Both methods showed robustness to lossy compression to a certain degree – if the compression was set to a high level it was impossible to embed information without noticeable distortions. Yet high levels of compression itself lead to noticeable distortions that, in the authors' opinion, make the methods robust against compression. Dubai and Haar methods were also able to embed with higher levels of compression (except the flags.mp4 file), but produced noticeable distortions on the video in order for the watermark to be readable.

Another type of attacks is geometric attacks [17] – three types of which were tested at first – cropping, logo overlay and both at once. An example is shown in Fig. 7.

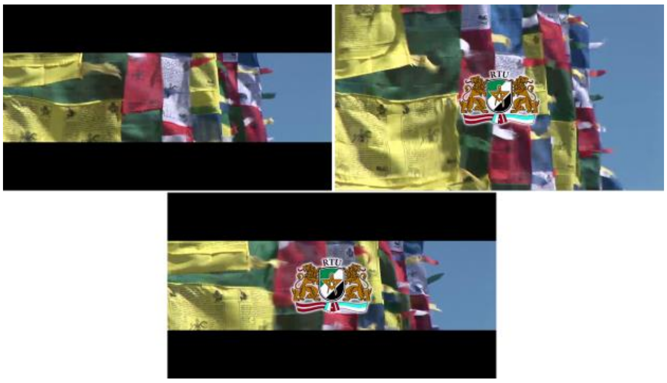


Fig. 7. Examples of attacks performed (Kaur adaptive method shown).

Both Kaur and Kothari methods were able to extract the embedded watermarks due to shuffling done previously – a loss of a portion of the frame resulted in data losses across the image – not in a single region, making it still readable (see Fig. 8).

TABLE VII

KAUR AND KOTHARI METHOD RESULTS (COMPRESSION WITH RATIO OF 1)

Container	Method	Δ	αFinal	αAverage	αMax	αMin	PSNR	Watermark identifiable
flags	Kaur	16	0.997	0.7468	0.9678	0.6944	35.61	Yes
flowers		8	0.9958	0.8314	0.9856	0.6885	39.38	Yes
riga		4	0.996	0.8973	0.9541	0.7812	40.13	Yes
flags	Kaur Adaptive	16	0.9963	0.7392	0.9584	0.6726	35.78	Yes
flowers		12	0.9919	0.8127	0.9733	0.6844	39.43	Yes
riga		6	0.9911	0.839	0.9253	0.7122	40.17	Yes
flags	Kothari	48	0.9954	0.8321	0.9967	0.7514	35.19	Yes
flowers		24	0.9969	0.9083	0.9952	0.7758	40.2	Yes
riga		12	0.9957	0.9341	0.9746	0.8478	43.05	Yes
flags	Kothari Adaptive	48	0.9949	0.8254	0.9965	0.7435	35.56	Yes
flowers		40	0.9925	0.9183	0.9932	0.7709	40.01	Yes
riga		20	0.9941	0.9063	0.9643	0.781	43.23	Yes

TABLE VIII

KAUR AND KOTHARI METHOD RESULTS (COMPRESSION WITH RATIO OF 0.5)

Container	Method	Δ	αFinal	αAverage	αMax	αMin	PSNR	Watermark identifiable
flags	Kaur	40	0.9798	0.7484	0.9931	0.6862	31.43	Partially
flowers		16	0.9995	0.8446	0.993	0.6696	37.41	Yes
riga		8	0.9953	0.9166	0.9656	0.733	39.41	Yes
flags	Kaur Adaptive	40	0.977	0.7428	0.9899	0.6696	31.7	Partially
flowers		24	0.9974	0.8219	0.9794	0.6821	37.48	Yes
riga		16	0.9944	0.9179	0.9668	0.737	39.36	Yes
flags	Kothari	64	0.9278	0.7719	0.9727	0.7242	32.47	Partially
flowers		32	0.9912	0.854	0.9895	0.7241	38.53	Yes
riga		20	0.9958	0.9439	0.986	0.834	41.31	Yes
flags	Kothari Adaptive	64	0.9091	0.758	0.9698	0.716	32.74	Partially
flowers		64	0.993	0.8897	0.9883	0.7301	37.47	Yes
riga		32	0.9923	0.9191	0.969	0.7616	41.79	Yes

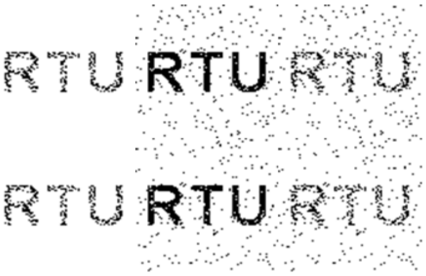


Fig. 8. Examples of watermarks extracted after geometric attacks with Kaur (top) and Kothari (bottom) methods – attacks (left to right): cropping, logo overlay, both

Table IX shows the comparison of results of both adaptive methods after attacks – they performed similarly, with Kothari method taking a slight lead when compared based on the change of αFinal before performing the attacks.

TABLE IX

RESULTS AFTER GEOMETRICAL ATTACKS

Attack	Method	Δ	αFinal	αAverage	αMax	αMin	αFinal change
Crop	Kaur Adaptive	6	0.9514	0.9256	0.9419	0.8902	−4.67 %
Logo overlay		6	0.9696	0.7854	0.9041	0.7399	−2.85 %
Both		6	0.9241	0.9013	0.9169	0.866	−7.40 %
Crop	Kothari Adaptive	16	0.9507	0.9268	0.9453	0.897	−4.37 %
Logo overlay		16	0.9675	0.81	0.9195	0.7694	−2.68 %
Both		16	0.9259	0.9048	0.9201	0.8772	−6.86 %

As a next step various rotation, stretching and scaling attacks were performed. None of the selected methods were able to perform well under these attacks (only Dubai method showed some resistance to a small degree of rotation – less than 0.2 degrees). This is a usual problem for both image steganography and video steganography methods [5], [17]. Possible solutions include embedding using Fourier-Merlin transformation [17] as well as trying to reverse the effects of the geometrical attacks during extraction [18]. These solutions are not perfect and are outside the scope of this paper.

Video steganography methods used in experiments showed the ability to successfully embed invisible watermarks; some showed robustness to compression and some degree of geometrical attacks. The authors were able to use Kothari and Kaur methods to embed invisible watermark into a video file, upload it to Youtube and Facebook and extract the data after downloading the video file back from the servers. Yet PSNR values were not enough to judge the fidelity of the video – to see how suspicious the video with embedded data was to a viewer a survey was performed.

VIII. Survey Results

To measure the impact of the noise added by the embedding process a survey was performed with 17 participants. Participants were given the original compressed video container of the combined video and 7 with watermarks embedded into them with different video steganography methods. The participant was to measure the visual video quality compared to the original and rate it on the scale of 0% (worst) to 100% (best). The explanation of the embedding process was given. The participants were also asked to describe the quality degradation that was visible. The summary of the survey is shown in Table X.

Results show that Kaur and Kothari methods performed the best with the given video container. Adaptive methods showed improvements for both methods – the average rating was higher and the MSE dropped for Kaur adaptive method. Haar method performed poorly even with low settings.

While analysing the description of video quality degradation, it was noted that some participants focused on colours, some – on visual artifacts. Some looked for artifacts in the background, some – in the moving parts of the image.

Only 4 participants noticed degradation of quality with Kaur method – small dots or a “net of dots”. The adaptive variant had 6 participants that noticed the degradation – one of the participants described issues that were in the original file. All the 4 participants who noticed the issues in the regular Kaur method noted that the degradation was less visible in the adaptive container.

For Kothari method results were similar – only 4 participants noticed any kind of degradation, the same 4 people found the same degradation in the adaptive variant and noted that it was less visible than the regular one.

To sum up, both Kaur and Kothari methods showed to be usable – adaptive methods demonstrated improvement for the both of these methods. As the research featured only 17 participants and they were not watching the videos in the same conditions (monitor size and configuration, lighting, distance to the screen etc.) the effects should be further researched in the future.

IX. Related Studies

Video steganography is a less researched field than image steganography, yet many embedding algorithms exist. Most of the methods do not have implementation publicly available, which leads to difficulties of comparing them. Some video steganography methods are developed for uncompressed containers (see [19]) and are not applicable to the internet. Methods that support compressed video containers use various transformations – usually DCT (see [13], [20]) or DWT (see [21]), while some latest methods use Fourier-Merlin transformation [17] or TPVD [6].

Most of the comparisons of steganographic algorithms (both image and video) feature the comparison of two algorithms with measurements and examples given (see [22]), usually between the new or improved algorithm and the algorithm that has been improved or some other “classic” algorithm. Comparisons that feature more methods are mostly theoretical (see [5]). A unified approach for practical comparison and measurement of video steganography algorithms as well as a framework for it is still missing.

TABLE X

Survey Result Summary (Embedding Strength Δ Given in Parentheses)

	Dubai (16)	Haar (2)	Kaur (12)	Kaur Adaptive (16)	Kothari (24)	Kothari Adaptive (32)
Average rating (%)	**89.58824**	**49.23529**	**91.11765**	**94.23529**	**95.41176**	**95.94118**
MSE	202.0069	117.8317	155.4882	15.53412	12.46797	11.77578
RMSE	14.21291	10.85503	12.46949	3.941335	3.531002	3.431585
Median	95	50	100	99	100	100
Rank (Delphi Method)	**5**	**6**	**4**	**3**	**1**	**1**

Other authors have also considered adaptive approaches, see [23] for an approach based on motion vectors and [24] for an approach based on regions of interest as examples. Facebook has also been selected as a possible channel for distributing stegocontainers in [7], which had little success with video steganography.

Some authors discuss ideas that are meant to improve the robustness of steganographic methods – embedding eigenvectors into the frame [25] or reversing the effects of geometrical attacks [18].

Steganalysis – the detection of the usage of steganography – is also active for the video steganograpy field, yet the total number of studies is quite low (see [26] and [27]).

X. Conclusion

The paper has reviewed the main approaches of video steganography, defined a set of criteria for a video steganography embedding algorithm to be used for uploading a video with hidden information to various online services. Four methods have been selected and compared based on their embedding efficiency and fidelity. Robustness against various attacks has been measured.

With the goal of decreasing the noise introduced by the steganographic process, yet keeping the fidelity level high, an improvement of two methods has been suggested. The improvement is based on adaptive embedding that takes into account the amount of movement in the video during the embedding process. The effectiveness of this approach as well as other methods has been examined both experimentally and by conducting a survey.

The adaptive method proposed is only an example of possible improvements – not only the movement and monotony in the frames is to be considered – various colours (blue, skin colour), transition effects and other factors are to be considered during embedding. Also embedding should be performed differently in separate scenes as to lower the chances of discovery using statistical methods. All of this brings a way for intelligent embedding methods that embed in a video based on its contents. Unfortunately, this approach requires far more processing power than regular ones and requires more research. The paper has proven, both with experiments and a survey, that video steganography can be used to unnoticeably embed hidden data into video containers that can be uploaded to various online video services and the data can still be extracted afterwards.

Further research will concentrate on improvement of the adaptive approach as well as extending the experiment from simple video fragments to longer and more complex videos with real viewers. Another direction for further research is the analysis of usability of video steganography from the side of the possible user or the client – digital media authors and copyright holders as well as persons that require a channel for hidden communication.

References

[1] E. Zielinska, W. Mazurczyk and K. Szczypiorski, “Trends in Steganography,” *Communications of the ACM*, no. 03, issue 57, 2014, pp. 86–95. http://dx.doi.org/10.1145/2566590.2566610

[2] I. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker, *Digital Watermarking and Steganography*, 2nd Ed., Burlington: Morgan Kaufmann, 2007.

[3] G. Konahovich and A. Puzirenko, *Digital Steganography, Theory and Practice – Компьютерная стеганография, теория и практика.* Kiev: MK-Press, 2006. (in Russian)

[4] A. K. Al-Frajat, H. A. Jalab, Z. M. Kasirun, A. A. Zaiden and B. B. Zaiden, “Hiding Data in Video File: An Overview,” *J. of Applied Sciences*, no. 10, pp. 1644–1649, June 2010. http://dx.doi.org/10.3923/jas.2010.1644.1649

[5] M. M. Sadek, A. S. Khalifa and M. G. M. Mostafa, “Video steganography: a comprehensive review,” *Multimedia Tools and Applications*, pp. 1–32, 2014. http://dx.doi.org/10.1007/s11042-014-1952-z

[6] A. P. Sherly, P. P. Amritha, “A Compressed Video Steganography using TPVD,” *Int. J. of Database Management Systems (IJDMS)*, vol. 2, no. 3, pp. 67–80, August 2010. http://dx.doi.org/10.5121/ijdms.2010.2307

[7] N. D. Amsden and L. Chen, “Analysis of Facebook Steganographic Capabilities,” in *Int. Conf. on Computing, Networking and Communications, Communications and Information Security Symposium*, 2015, pp. 67–71. http://dx.doi.org/10.1109/iccnc.2015.7069317

[8] “MSU Video StegoVideo tool and plugin,” March 2011. [Online]. Available: http://www.compression.ru/video/stego_video/index_en.html. [Accessed: October 3, 2015].

[9] “OpenPuff tool,” 2015. [Online]. Available: http://embeddedsw.net/OpenPuff_Steganography_Home.html. [Accessed: October 3, 2015].

[10] “RT Steganography in Video Streaming tool,” July 2013. [Online]. Available: http://sourceforge.net/projects/rtstegvideo. [Accessed: October 3, 2015].

[11] “Steganosaurus tool,” April 2013. [Online]. Available: http://steganosaur.us. [Accessed: October 3, 2015].

[12] B. Kaur, A. Kaur and J. Singh, “Steganographic Approach for Hiding Image in DCT Domain,” *Int. J. of Advances in Engineering & Technology*, vol. 1, issue 3, pp. 72–78, July 2011.

[13] A. M. Kothari and V. V. Dwivedi, “Performance Analysis of Digital Video Watermarking using Discrete Cosine Transform,” *Int. J. of Electrical and Computer Engineering Systems*, vol. 2, no. 1, pp. 11–16, 2011.

[14] S. AL-Mansoori and A. Kunhu, “Robust Watermarking Technique based on DCT to Protect the Ownership of DubaiSat-1 Images agains Attacks,” *IJCSNS Int. J. of Computer Science and Network Security*, vol. 12, no. 6, June 2012.

[15] K. R. Chetan and K. Raghavendra, “DWT Based Blind Digital Video Watermarking Scheme for Video Authentication,” *Int. J. of Computer Applications*, vol. 4, no. 10, pp. 19–26, Augusts 2010. http://dx.doi.org/10.5120/863-1213

[16] “Xuggler,” 2011. [Online]. Available: http://www.xuggle.com/xuggler. [Accessed: October 3, 2015].

[17] Dai-Kyung Hyun and Heung-Kyu Lee, “A Low-Complexity Mobile Watermarking Scheme Resisting Scale Distortions,” *J. of Computer and Communications*, vol. 2, pp. 77–81, 2014. http://dx.doi.org/10.4236/jcc.2014.24011

[18] M. Kutter, F. Jordan and F. Bossen, “Digital Signature of Color Images using Amplitude Modulation,” January 1997. [Online]. Available: http://www.alpvision.com/pdf/3022-51.pdf [Accessed: Oct. 11, 2015].

[19] A. Swathi and S.A.K. Jilani, “Video Steganography by LSB Substitution Using Different Polynomial Equations,” *Int. J. Of Computational Engineering Research*, vol. 2, issue 5, pp. 1620–1623, September 2012.

[20] S. Gujjunoori and B. B. Amberker, “Reversible Data Embedding for MPEG-4 Video Using Middle Frequency DCT Coefficients,” *J. of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 3, pp. 408–419, July 2014.

[21] R. J. Mstafa and K. M. Elleithy, “A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11),” in *Wireless Telecommunications Symposium (WTS)*, 2015, pp. 1–8. http://dx.doi.org/10.1109/wts.2015.7117257

[22] N. Chaturvedi and S. J. Basha, “Comparison of Digital Image watermarking Methods DWT & DWT-DCT on the Basis of PSNR,” *Int.*

J. of Innovative Research in Science, Engineering and Technology, vol. 1, no. 2, pp. 147–153, December 2012.

[23] J. Mansouri and M. Khademi, “An adaptive scheme for compressed video steganography using temporal and spatial features of the video signal,” *Int. J. of Imaging Systems and Technology*, vol. 19, issue 4, pp. 306–315, December 2009. http://dx.doi.org/10.1002/ima.20207

[24] S. Khupse and N. N. Patil, "An adaptive steganography technique for videos using Steganoflage," in *Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2014, pp.811-815.

[25] S. A. Hosseini-pour, M. Soleimanpour and H. Nezamabadi-pour, “A novel approach to secure image based steganography by using eigenvalue and eigenvector principles,” in *2013 21st Iranian Conference on Electrical Engineering*, ICEE, 2013, pp. 1–5. http://dx.doi.org/10.1109/IranianCEE.2013.6599635

[26] U. Budhia, D. Kundur and T. Zourntos, “Digital Video Steganalysis Exploiting Statistical Visibility in the Temporal Domain,” in *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 502–516, December 2006. http://dx.doi.org/10.1109/TIFS.2006.885020

[27] K. Bryan, “Video Steganalysis for Digital Forensics Investigation,” PhD thesis, University of Rhode Island, Kingston, US, 2013.

David Griberman, Mg. sc. ing. (2015), Bc. sc. ing. (2013) – Riga Technical University (RTU). Diploma with distinction: Master of engineering science in computer systems.

He is an Oracle Programmer at IT company Lattelecom Technology. Fields of interest: steganography, hidden communications, copyright protection of digital media.

E-mail: st15@griberman.com

Pavel Rusakov was born in Riga, Latvia, in 1972. The degrees obtained: Dr. sc. ing. (1998), Mg. sc. ing. (1995), Bc. sc. ing. (1993) – Riga Technical University (RTU). Diploma with distinction: Mg. sc. ing.

He is an Associated Professor at the Institute of Applied Computer Systems, RTU. He is the Head of Laboratory, responsible for the Professional Bachelor and Master Studies at the Department of Applied Computer Science. Field of interest: computer science. Special interest: programming paradigms, object-oriented approach to systems development, parallel computing, web technologies, distributed systems, computer graphics, and protection of information.

E-mail: Pavels.Rusakovs@cs.rtu.lv