FAST IMPLEMENTATION OF DIGITAL WATERMARKING SCHEMES BASED ON ARNOLD AND DISCRETE WAVELET TRANSFORMS

A. G. Zotin [1], A. V. Proskurin [1]

[1] Institute of Computer Science and Telecommunications, Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russian Federation - zotin@sibsau.ru, proskurin.av.wof@gmail.com

Commission II, WG II/5

KEY WORDS: Digital Watermarking, Arnold Transform, Discrete Wavelet Transform, Lookup Table.

ABSTRACT:

In recent years, digital watermarking of photo and video materials has become more and more important in connection with the transmission of multimedia data over unsecured communication channels. Digital watermarking allows to reduce the amount of transmitted information and to protect embedded metadata. Improving robustness and security of embedded data increases computational costs, which obstruct usage of digital watermarks in mobile devices. In this research, we propose a number of improvements to the digital watermarking process based on Arnold and discrete wavelet transforms to reduce the computational cost. Considering the watermark as a linear sequence of pixels allows us to speed up its processing. The two-dimensional lookup table allows performing an Arnold transform in constant time regardless of the number of iterations. Number of iteration for each block of watermark is determined using hash function applied to the secret key. Also, the structure of the lookup table is proposed to accelerate the embedding of watermark. This table allows to determine the frequency coefficients for embedding based on the key hash code. Proposed improvements allow to speed up the watermark preparation by an average 14 times and the overall embedding process by 1.22 times for 1920×1080 images.

1. INTRODUCTION

Due to the active development of mobile and network technologies, an increasing amount of multimedia content in the form of images and videos are transmitted through unsecured communication channels. One of the ways to protect copyright and transfer confidential information is to embed a digital watermark. In this case, the secret information usually takes the form of a small image (watermark) and is hidden inside the main image (host) with minimal visual distortion of the latter. The watermark can be retrieved back to its original form, which is used for ownership verification. Thus, the digital watermark and the algorithm for its embedding must have the following properties (Begum, Uddin, 2020; Verma, Jha, 2015):

- Imperceptibility – a watermark embedding should not make obvious visual distortions of the host and hidden information should not be visible to humans.
- Robustness – common attacks on the host, such as JPEG-compression, filtering and cropping, should not lead to significant distortion of the watermark or make it difficult to extract.
- Capacity – it is necessary to embed as much hidden information into the host as possible, adding more information about author or duplicating it to increase probability of successful extraction.
- Computational cost – a mobile device should be able to embed a watermark in a high-resolution image within a reasonable time frame.

For watermark embedding spatial or frequency (transform) methods are used. Spatial methods are based on a direct change in the intensity of pixels in a selected area of the host image. The most famous spatial methods are the least significant bit (LSB) method and its modifications (Abraham, Paul, 2019; Feng et al., 2020) as well as the intermediate significant bits and its modification (Mohammed et al., 2014; Zeki, Manaf, 2009). These methods are easy to implement and allow to embed a large amount of information. However, watermarks embedded in this way are easily detected by computer analysis or visually and unable to effectively resist most of attacks.

Frequency-based methods make use of various transformations of host image to the frequency domain (Discrete Fourier Transform (Gaata, 2016), Discrete Cosine Transform (DCT) (Roy et al., 2017), Discrete Wavelet Transform (DWT) and its modifications (Khare, Srivastava, 2019; Tan et al, 2020), Singular Value Decomposition (SVD) (Li et al., 2016)). They conduct watermark embedding in the “mid-frequency” components, which simultaneously improves imperceptibility and robustness of the watermark. Negative side of these methods is significantly smaller data payload and high computational costs. However, due to its resistance to many types of attacks, watermarking based on frequency methods has become widespread in recent years (Zotin et al., 2020).

In general, process of watermark embedding also contains stage of confidential information preparation (formation of a watermark). The main purpose of preparation stage is to transform the embedding information in the sequence of bits, which increase imperceptibility, robustness and security of data transmission. Favorskaya and Zotin (2020) proposed to use textual information embedding algorithm based on Code 128 barcoding and DWT. This increases probability of correct information retrieval even in case of severe damage to the watermark.

Additionally, the watermark is scrambled using Arnold transform (Li et al., 2013) for better imperceptibility. This scrambling algorithm is based on the iterative change of the

image pixel positions through a matrix transformation, which gives visual effect of disorder. This increases resistance to attacks such as cropping, noise, compression and filtering, and also makes it difficult to detect watermark using computer analysis. However, Arnold transform is computationally expensive due to several iterations over elements of the two-dimensional matrix. Also in most cases, the same number of iterations is used for all parts of the watermark, which reduces security.

In this paper, we propose several improvements to the aforementioned method of digital watermarking to reduce computational cost and increase data encryption. Our method of source data preparation considers dividing entire watermark into sequences of bits and processes them linearly. Scrambling is done using lookup tables to achieve constant runtime. To improve information imperceptibility and security, each block is transformed using a different number of iterations, which is determined by MD5 hash generated for owner-specified key. Additionally, at the stage of embedding, lookup tables are used for fast implementation of DWT and selection of coefficients used for embedding bits of watermark in the region, obtained by performing two levels of wavelet decomposition.

The rest of the paper is organized as follows. In Section 2, the related works in digital watermarking schemes are briefly reviewed. The description of our proposed framework is presented in Section 3. Some empirical results and discussions are demonstrated in Section 4. Finally, conclusions are drawn in Section 5.

2. RELATED WORKS

Over the past decades, a large number of digital watermarking methods have been proposed. Initially, spatial-based methods, such as LSB, were widespread. In these methods, LSB of randomly chosen pixels can be altered to hide the most significant bit (MSB) of another. The watermark is inserted into the least significant bits of the host image and can be extracted in the same way. This is easy to implement and has minimal impact on quality of the host image. However, these methods are too fragile and can be affected by undesirable noise, cropping, lossy compression and so on. Because of this, robust watermarking schemes have been implemented using DCT.

The method proposed in (Kitanovski et al., 2005) divided an image into $P \times P$ blocks, after which performed a block-based DCT transform in each blocks and embedded in the low-frequency components one watermark bit per block. A watermark was generated using an image hash as a key. This method demonstrated high robustness in image authentication, but cause severe visual distortions of the host image. Pun (2009) introduced a method, which embeds 4096 bits of information in 512×512 pixels image using adaptive quantization to select 12 representative DCT coefficients for embedding. This method was robust against Gaussian low pass filter and JPEG compression.

Roy and Pal (2017) proposed a color multiple watermarking method based on DCT and repetition code for ownership protection and validation. In this method, a binary bit of several watermarks were embedded into green and blue color components of the host image. For this purpose, DCT blocks were zigzag scanned after which some middle significant DCT-coefficients were modified using repetition code. The system demonstrated better imperceptibility and robustness and generated a higher PSNR value by eliminating blocking artifacts. However, the system exhibited high computational complexity. The method proposed by Loan et al. (2018) used chaotic encryption and DCT for grayscale and color images watermarking. The watermark bit was embedded by modifying difference between DCT coefficients of adjacent blocks. To add another layer of security, Arnold transforms along with a chaotic map were used at this time. The results demonstrated the robustness of the system against common image processing operations.

Tsai et al. (2017) proposed image watermarking method based on the fast DCT algorithm for implementation in digital signal processor. The keys in watermarking processed include four frequency coefficients in DCT, two random permutation vectors and a quantization matrix for normalizing the watermark and the host image. The fast DCT algorithm reduced the complexity of two-dimensional image transformation, but watermark embedding still took 0.33 sec for 256×256 pixels image.

Additionally, many studies have been carried out on the authentication of images using DWT. For example, Haribabu et al. (2016) proposed method based on the wavelet transform in HSI color space for protecting copyright holder information. For intensity components of host image and watermarks 1-level wavelet coefficients were generated. For embedding, each 8×8 block of both images was compared and scaled with a scaling factor α, after which host image inverse transformed. The simulation results demonstrated that this scheme is more robust against noise, but original image is needed for extraction.

The method proposed in (Jia et al., 2017) extracted watermark from watermarked image without requirement of original host image or original watermark image. This method was based on DWT and QR decomposition. Initially, each component of the color host image was transformed by 1-level DWT and further divided to 4×4 non-overlapping pixel blocks. Then, each selected pixel block is decomposed by QR decomposition and the first row elements in the matrix R was quantified for embedding the watermark information. Method had better robustness against noise, image compression, cropping and filtering, but high computational complexity.

In recent years, more researchers combine different transform methods for image watermarking. For example, in paper (Khare, Srivastava, 2019) authors proposed new image watermarking method, which utilizes properties of homomorphic transform, redundant DWT, Arnold transform along with SVD. Redundant DWT was performed on host image to achieve LL subband, which further decomposed into illumination and reflectance components by homomorphic transform. In order to strengthen security of proposed scheme, Arnold transform was used to scramble watermark embedded with singular values of reflectance component, which are obtained by applying SVD to it. This method demonstrated excellent imperceptibility and good robustness.

From the above studies, we can conclude that transform-based techniques are robust against common image processing operations. However these methods require huge amount of calculation and subsequent researches, which mostly aim at improving robustness and imperceptibility, increase computational cost. This makes transform-based methods difficult to use on mobile devices.

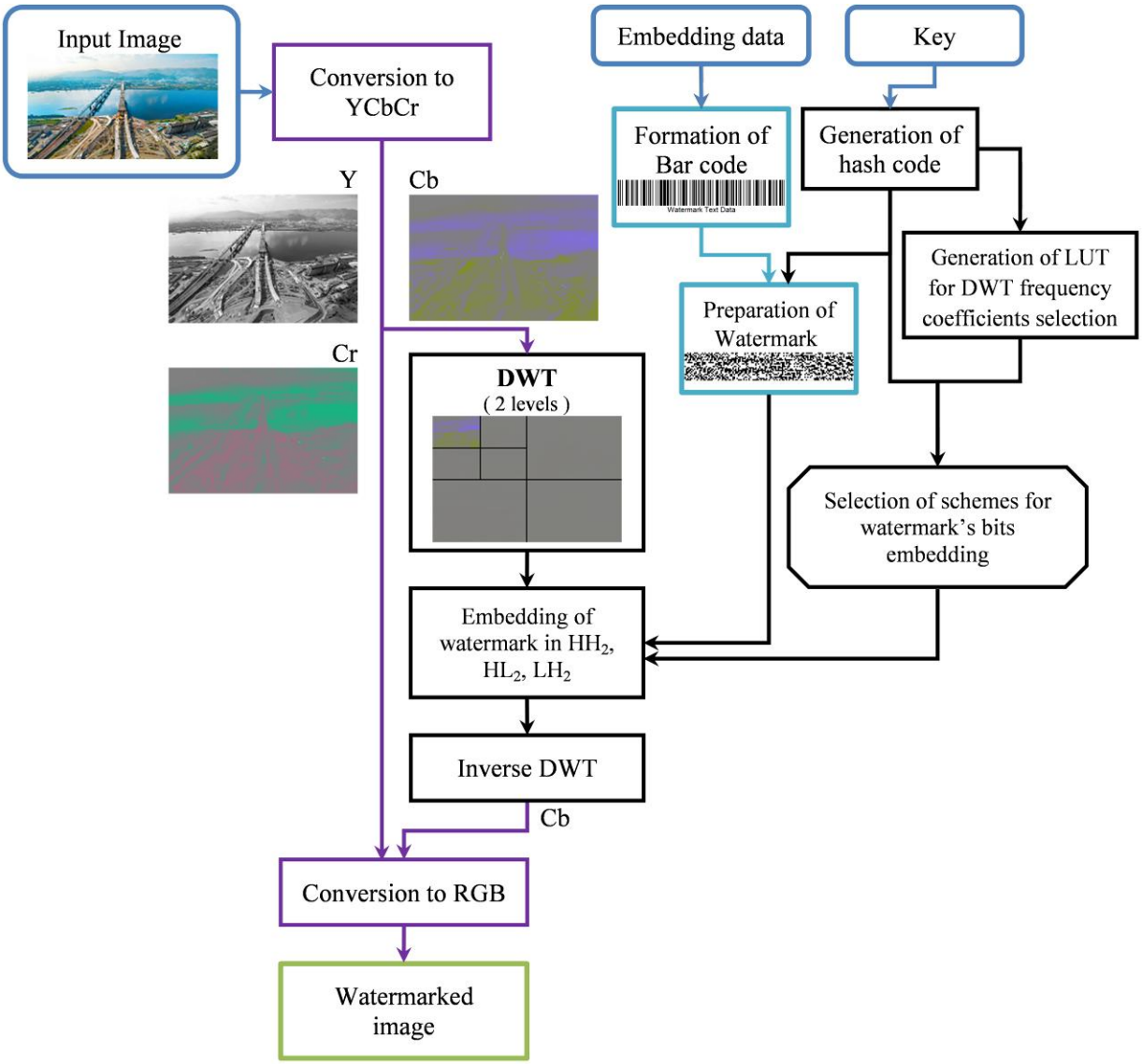

Figure 1. Flow-chart of the proposed watermark embedding method.

3. PROPOSED METHOD

We propose a number of computational improvements in watermark preparation and embedding steps based on the method described in (Favorskaya, Zotin, 2020). This method is referred as basic in the rest of the paper. The flow-chart of proposed method is presented in Figure 1. The method consists of 6 stages: preprocessing of text information (formation of watermark bit's structure), conversion of a host image to the YCbCr color model, 2-level DWT of Cb color channel, watermark embedding using adaptation of the Koch-Zhao algorithm for frequency coefficients modification, inverse DWT, and conversion of a watermarked image to the RGB color model. Improvements allow to reduce overall computational cost, strengthen imperceptibility and encryption of embedded data. In the following subsections, we describe preprocessing and embedding steps in details.

3.1 Preprocessing

The purpose of watermark preparation is to convert embedded information into such a sequence of bits that would improve reliability and security of data transmission. In the basic method, text information is encoded with a Code 128 barcode to increase probability of correct information reading even if the watermark is severely damaged. In the next step, the basic method splits barcode into patches of 32×16 pixels and forms one block from every two patches. Such scheme is easily perceived by a person, however, block formed in this way carries only part of the embedded information. In the case of host distortion, recovery will largely depend on the cyclical repetition of the embedded watermark.

The proposed modification assumes recording of partial information about the entire barcode in each block. For this purpose, barcode is interpreted linearly. If width of barcode data is less than 1024 pixels, then one block of 32×32 pixels includes several fragments at once with cyclic repetition. This increases probability of correct barcode reconstruction. The scheme of linear reading of the digital watermark is shown in Figure 2.

In the next step, Arnold transform is applied for better data hiding. This transform applies only to square images, however,

it is also possible to apply Arnold transform to one-dimensional representations in two-dimensional interpretation.

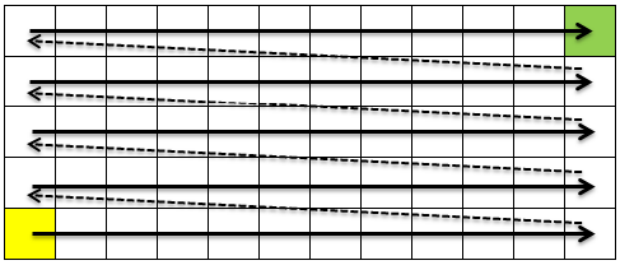

Figure 2. Scheme of a watermark linear reading.

For a $N \times N$ block, Arnold transform changes coordinates of block element to new coordinates according to the expression:

$$\begin{bmatrix} X_{new} \\ Y_{new} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \bmod N, \quad (1)$$

where X, Y = old coordinates of the block element
X_{new}, Y_{new} = new coordinates of the block element

The key feature of Arnold transform is that after a certain number of iterations original block is obtained. Table 1 shows such number of iterations for typical block sizes. An example of Arnold transform for a 32×32 pixel block is demonstrated in Figure 3.

Block size	Iterations	Block size	Iterations
8×8	6	72×72	14
12×12	12	80×80	60
16×16	12	88×88	30
24×24	12	96×96	24
32×32	24	104×104	42
40×40	30	112×112	24
48×48	12	120×120	60
56×56	24	128×128	96
64×64	48	256×256	192

Table 1. The number of iterations for different block sizes, after which original block will be obtained.

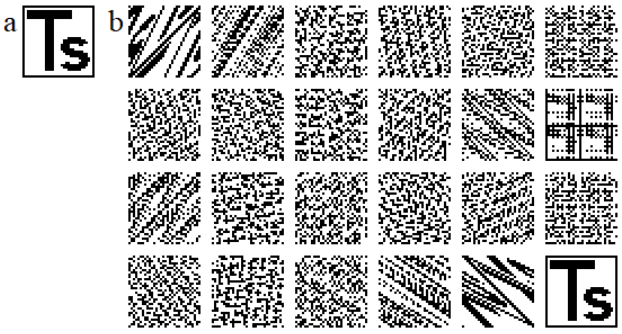

Figure 3. The example of Arnold transform for 32×32 block: a) original block, b) from left to right and from top to bottom – 24 iterations of transform.

Direct application of Arnold transform leads to high computational costs due to multiple iterations over a two-dimensional matrix. In this regard, we propose to use lookup tables. Let's rewrite expression (1) in the following form:

$$\begin{aligned} X_{new} &= (X + Y) \bmod N \\ Y_{new} &= (X + 2 \cdot Y) \bmod N \end{aligned} \quad (2)$$

It is necessary to take into account linear interpretation of the block for lookup table generation. In this case, formation of the lookup table $A1D$ used for one scrambling iteration is performed by equation (3), which uses equation (2).

$$A1D[Y \cdot N + X] = Y_{new} \cdot N + X_{new} \quad (3)$$

Additionally, the lookup table $A2D$ is formed not to repeat application of the $A1D$ many times. With this table, Arnold transform is performed in constant time, regardless of the selected number of iterations. The first level in the $A2D$ means the iteration number and the second level – transform parameters (similar to $A1D$). Values at the first level ($A2D[1]$) are equal to values from table $A1D$. The subsequent levels are calculated according to equation:

$$A2D[i][p] = A2D[i-1][(A2D[1][p])], \quad (4)$$

where i = number of the current transform iteration
p = position of the transform parameter

Each block of watermark is transformed with its own number of iterations. In the basic method, its determination is based on a secret key, which was processed by a set of complex rules with many conditions due to the presence of arbitrary symbols. This situation can be avoided by using a limited set of symbols, but in this case it will be easier to brute force secret key in order to maliciously extract watermark and read hidden information. Also, numbers of iterations were regularly repeated if the key was small.

To solve these problems, we propose to form the lookup table *AAlter*, which will determine the number of transform iterations used for the block. To generate it, parameters of the hash code obtained by applying MD5, SHA-256, SHA-384 or SHA-512 algorithms to the secret key are used. Examples of different hash code sequences are given in Table 2.

Hash function	Hash code
Raw key	Test@Key#Watermark
MD5	B17740EE08AED1A996328C3081A8537C
SHA-256	8140778612769DBAB2A6A874B535C0AD7 6206F41C18CAEC54A2BD8492B80B122
SHA-384	F4C9ABEDB1DD8940E2751E802F82FE5CE 376D919E72218F684571F97E5F47F3E6293 2EF8BA81D96FCB7D1693D10450DD
SHA-512	446AB01A554D7B25143CB1AF94CD8084F 5039EC28072FA406070410DC179F7603E5 AEA09B7B74E1B75A4F31F3962774961A2 B5DB5D7E16B3BA523EE39A6E7C04

Table 2. Examples of different hash codes for key.

Symbol of hash code takes a value of digits 0-9 and Latin letters A-F, which together can be interpreted as numbers 0-15. *AAlter* table is filled depending on the set of valid numbers of iterations. For example, the number of iterations is determined using basis (first symbol of the code) and additional offset (residuals from sum of 1-3 subsequent symbols) for

corresponding watermark block. In the case of using the MD5 code, this allows to set the number of iterations for 16, 10 and 8 blocks, respectively. To embed more information, we can either loop through the hash code or use a different hash function. These actions encrypt transmitted message and increase security.

An example of textual information transform using basic and proposed methods is shown in Figure 4.

Figure 4. The example of textual information presentation: a) original text, b) barcode presentation, c) shrunk barcode (basic method), d) shrunk barcode after Arnold transform (15 iterations), e) linear interpretation of barcode (proposed method), f) linear interpretation after Arnold transform (use of secret key).

3.2 Embedding

The watermark embedding algorithm has not changed significantly compared to the basic method. Host image, watermark bit sequence obtained at the preparation stage and owner-specified secret key are used as input data. The following steps are applied to this data:

1. Convert the host image from RGB to YCbCr color space.
2. Apply direct 2-level Haar wavelet transform to Cb color channel.
3. Embed textual watermark using a modified Koch-Zhao algorithm in HH_2, HL_2 and LH_2 regions.
4. Apply inverse 2-level Haar wavelet transform.
5. Convert the watermarked image from YCbCr to RGB color space.

The changes affect three parts: calculation of direct and inverse DWT, and also determining scheme for watermark's bits embedding by a modified Koch-Zhao algorithm. To speed up DWT, we use integer calculations and a lookup tables generated to calculate DWT coefficient values. To determine bits embedding scheme, we use a linear interpretation of the frequency domain and the lookup table *WMX*, which built using hash codes of the secret key and frequency domain identifier.

The embedding of watermark's bit is performed using 2-level DWT frequency coefficients. The selection of coefficient pair for embedding is carried out from four candidates (P_1 P_2 P_3 P_4). Therefore, 12 different combinations are possible. The initial *WMX* basis is filled based on these 12 combinations. For this purpose, a cyclic shift is performed based on hash code obtained for the secret key and selected frequency domain. The remainder of dividing first hash code symbol by 12 is used to determine shift of the base part. Shift type is determined based on the remainder of dividing second symbol by 3. In this case, 0 is interpreted as a shift of all elements, 1 – only even elements and 2 – only odd elements.

Additionally, the *WMX* has been expanded to 16 elements with duplicate pairs for easy use of hash code symbols. The pairs definition is based on first four symbols of a MD5 hash code of the secret key. Examples of base part and complete table *WMX* formation are demonstrated in Figures 5 and 6, respectively. A sequential set of pairs used for bits embedding is determined based on the SHA-256, SHA-384 or SHA-512 codes, taking into account the iterative embedding relative to the image size.

MD5 hash for SecretKeyHH2 **B5A0CE6CE7E9D45FA87787249B2DF34E**
B mod 12 = 11
5 mod 3 = 2

Base Pair table	
1	{P1, P2}
2	{P1, P3}
3	{P1, P4}
4	{P2, P1}
5	{P2, P3}
6	{P2, P4}
7	{P3, P1}
8	{P3, P2}
9	{P3, P4}
10	{P4, P1}
11	{P4, P2}
12	{P4, P3}

Pair table (marked)	
1	{P1, P2}
2	{P1, P3}
3	{P1, P4}
4	{P2, P1}
5	{P2, P3}
6	{P2, P4}
7	{P3, P1}
8	{P3, P2}
9	{P3, P4}
10	{P4, P1}
11	{P4, P2}
12	{P4, P3}

Pair Table (Transformed)	
3	{P1, P4}
2	{P1, P3}
5	{P2, P3}
4	{P2, P1}
7	{P3, P1}
6	{P2, P4}
9	{P3, P4}
8	{P3, P2}
11	{P4, P2}
10	{P4, P1}
1	{P1, P2}
12	{P4, P3}

Figure 5. The example of *WMX* basis formation for key "SecretKey" and frequency domain HH_2.

The MD5 hash for SecretKey **0D734A1DC94FE5A914185F45197EA846**

Symbol		Answer
0	mod 12	0
D		1
7		7
3		3

Base Pair table	
1	{P1, P2}
2	{P1, P3}
3	{P1, P4}
4	{P2, P1}
5	{P2, P3}
6	{P2, P4}
7	{P3, P1}
8	{P3, P2}
9	{P3, P4}
10	{P4, P1}
11	{P4, P2}
12	{P4, P3}

WMX		
Base part	0	{P1, P4}
	1	{P1, P3}
	2	{P2, P3}
	3	{P2, P1}
	4	{P3, P1}
	5	{P2, P4}
	6	{P3, P4}
	7	{P3, P2}
	8	{P4, P2}
	9	{P4, P1}
	A	{P1, P2}
	B	{P4, P3}
Extension	C	{P4, P3}
	D	{P1, P2}
	E	{P3, P1}
	F	{P1, P4}

Figure 6. The example of *WMX* formation for key "SecretKey" and frequency domain HH_2.

3.3 Extraction

The watermark extraction algorithm assumes performing embedding steps in reverse order. The input data is a watermarked image and a secret key. The extraction scheme involves following steps:

1. Convert the watermarked image from RGB to YCbCr color space.
2. Apply direct 2-level Haar wavelet transform to Cb color channel.
3. Extract the textual watermark from HH_2, HL_2 and LH_2 regions.
   a. Create *WMX* table using secret key.
   b. Extract bits from HH_2, HL_2 and LH_2.

c. Transform extracted bit’s watermark in blocks, apply Arnold transform using required number of iterations.
d. Improve vertical strips and decode using Code-128.

As the output we receive textual data of embedded watermarks.

4. EXPERIMENTS AND RESULTS

For main experiments, the dataset obtained by Micro Aerial Vehicle (MAV) camera was used (András, Scaramuzza, 2017). The shooting was conducted in urban territory at a low altitude of 5-15 meters above the ground. Image examples from the used dataset are depicted in Figure 7. This dataset includes 81,169 images with high resolution (1920×1080) and log data from on-board Global Positioning System (GPS) receiver. Additionally, we use 100 images each of other resolutions: 1280×720, 2560×1440 and 3840×2160. All calculations were performed on single core of i7 4770 CPU.

Figure 7. Examples from MAV dataset: a) 00001.jpg, b) 20001.jpg, c) 40001.jpg, d) 60001.jpg.

Experiments can be divided into three parts:

1. Testing acceleration of Arnold transform.
2. Testing acceleration of DWT.
3. Testing overall acceleration of watermark embedding.

In the first experiment, we compared time of Arnold transform in the basic method and using the lookup tables *A1D* and *A2D*. For all time measurements 10,000 calculations were performed and average values are presented as the result. A visualization of the results for 32×32 blocks (24 iterations is the maximum) are demonstrated in Figure 8. The results show that use of table *A1D* allows to get an acceleration of 2-2.5 times, while table *A2D* allows to transform block in constant time. For 24 iterations the acceleration reaches 30 times. For larger blocks (and, accordingly, for more iterations) the speedup will be greater.

During second experiment, acceleration of the DWT from use of integer calculations and lookup tables was tested. All calculations were repeated 100 times, after which average values were found. The results for 1-level and 2-level DWT using images of different resolutions are presented in Tables 3 and 4, respectively. We found that average acceleration is 5.7% for 1-level DWT and 4.2% for 2-level DWT.

In the last experiment, overall acceleration of watermarking was tested using the dataset obtained by MAV. Log data from a GPS receiver was used as the transmitted textual information. The following string was used as a secret key: “Test@Key#Watermark”. During testing, we additionally calculated time spent at each step of watermark embedding. The results obtained for each image were averaged and presented in Table 5. It can be seen that preparation of watermark was accelerated by an average of 14.3 times and the total time was reduced by 1.22 times.

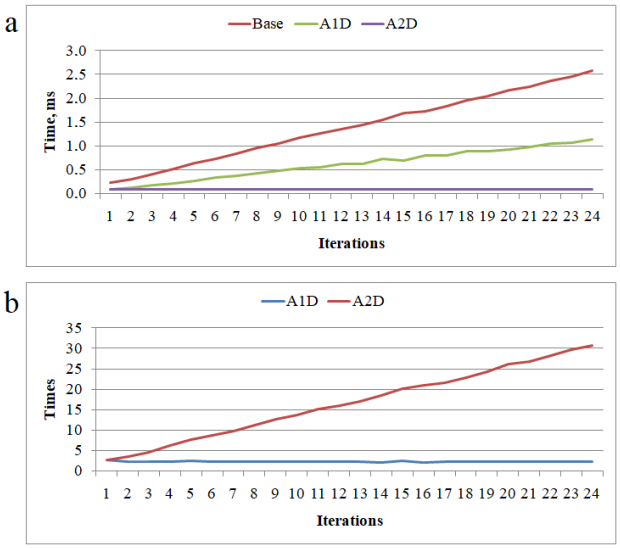

Figure 8. Comparison of Arnold transform calculation in basic method (Base) and using tables *A1D* and *A2D* for 32×32 blocks: a) calculation time, b) speedup.

Resolution	Base, ms	LUT, ms	Speedup, %
1280×720	5.11	4.82	5.7
1920×1080	11.58	10.89	5.9
2560×1440	20.37	19.11	6.2
3840×2160	43.81	41.56	5.1

Table 3. Comparison of 1-level DWT calculation in basic method (Base) and proposed lookup table (LUT).

Resolution	Base, ms	LUT, ms	Speedup, %
1280×720	6.03	5.78	4.1
1920×1080	13.26	12.69	4.3
2560×1440	23.69	22.75	4.0
3840×2160	50.96	48.76	4.3

Table 4. Comparison of 2-level DWT calculation in basic method (Base) and proposed lookup table (LUT).

Stage	Base, ms	Proposed, ms	Speedup, times
Preprocessing	12.63	0.88	14.31
RGB to YCbCr	15.80	15.80	1.00
Direct DWT	13.26	12.69	1.04
Embedding	1.29	1.01	1.28
Inverse DWT	16.61	16.15	1.03
YCbCr to RGB	12.79	12.76	1.00
Without color conversion	43.80	30.74	1.42
All process	72.40	59.31	1.22

Table 5. Comparison of basic and proposed methods.

5. CONCLUSIONS

In this study, we propose fast implementation of digital watermarking based on Arnold and discrete wavelet transforms. In order to reduce computational cost we use integer calculation and lookup tables for DWT, which gives 4-6% of speedup. Also, the proposed two-dimensional lookup table for Arnold transform allows to perform transformation in the constant time. It gives up to 30 times of speedup for a 32×32 pixels block. To strengthen security, we proposed an algorithm for determination of Arnold transform iterations based on the secret key hash code. The experiments show high efficiency of the proposed improvements when a full embedding process speeds up to 1.22 times for 1920×1080 images. This allows more efficient usage of watermarking technology in mobile devices. Further acceleration of watermark embedding depends on optimizing conversions between color spaces.

ACKNOWLEDGEMENTS

The reported study was funded by Russian Foundation for Basic Research to the research project no. 19-07-00047 a.

REFERENCES

Abraham J., Paul V., 2019. An Imperceptible Spatial Domain Color Image Watermarking Scheme. *Journal of King Saud University – Computer and Information Sciences*, 31(1), 125-133.

András M.L., Till C., Scaramuzza D., 2017. The Zurich urban micro aerial vehicle dataset. *International Journal of Robotics Research*, 36(3), 269-273.

Begum M, Uddin M.S., 2020. Digital Image Watermarking Techniques: A Review. *Information*, 11(2), 110.

Favorskaya M., Zotin A., 2020. Robust textual watermarking for high resolution videos based on Code-128 barcoding and DWT. *Procedia Computer Science*, 176, 1261-1270.

Feng B., Li X., Jie Y., Guo C., Fu H., 2020. A Novel Semi-fragile Digital Watermarking Scheme for Scrambled Image Authentication and Restoration. *Mobile Networks and Applications*, 25(12), 82-94.

Gaata M.T., 2016. An Efficient Image Watermarking Approach Based on Fourier Transform. *International Journal of Computer Applications*, 136, 8-11.

Haribabu M., Bindu C.H., Swamy K.V., 2016. A Secure & Invisible Image Watermarking Scheme Based on Wavelet Transform in HSI color space. *Procedia Computer Science*, 93, 462–468.

Jia S., Zhou Q., Zhou H., 2017. A Novel Color Image Watermarking Scheme Based on DWT and QR Decomposition. *Journal of Applied Science and Engineering*, 20, 193-200.

Kitanovski V., Taskovski D., Bogdanova S., 2005. Watermark Generation using Image-Dependent Key for Image Authentication. *EUROCON 2005 - The International Conference on "Computer as a Tool"*, 947-950. doi.org/10.1109/EURCON.2005.1630103.

Khare P., Srivastava V., 2019. Secure and Robust Image Watermarking Scheme Using Homomorphic Transform, SVD and Arnold Transform in RDWT Domain. *Advances in Electrical and Electronic Engineering*, 17(3), 343-351.

Li M., Liang T., He Y., 2013. Arnold transform based image scrambling method. *Proceedings of 3rd International Conference on Multimedia Technology (ICMT-13)*, 1302-1309.

Li Y., Wei M., Zhang F., Zhao J., 2016. A New Double Color Image Watermarking Algorithm Based on the SVD and Arnold Scrambling. *Journal of Applied Mathematics*, 2016, 9. doi.org/10.1155/2016/2497379.

Loan N.A., Hurrah N.N., Parah S.A., Lee J.W., Sheikh J.A., Bhat G.M., 2018. Secure and Robust Digital Image Watermarking Using Coefficient Differencing and Chaotic Encryption. *IEEE Access*, 6, 19876-19897. doi.org/10.1109/ACCESS.2018.2808172.

Mohammed G.N., Yasin A., Zeki A.M., 2014. Robust Image Watermarking Based on Dual Intermediate Significant Bit (DISB). *Proc. of the 6th International Conference on CSIT*, 18-22. doi.org/10.1109/CSIT.2014.6805973.

Pun C., 2009. High Capacity and Robust Digital Image Watermarking. *Fifth International Joint Conference on INC, IMS and IDC*, 1457-1461. doi.org/10.1109/NCM.2009.85.

Roy S., Pal A.K., 2017. A Blind DCT Based Color Watermarking Algorithm for Embedding Multiple Watermarks. *AEU - International Journal of Electronics and Communications*, 72, 149-161.

Tan L., He Y., Wu F., Zhang D., 2020. A Blind Watermarking Algorithm for Digital Image Based on DWT. *4th International Conference on Machine Vision and Information Technology (CMVIT 2020)*, 1518. doi.org/10.1088/1742-6596/1518/1/012068.

Tsai S.E., Liu K.C., Yang S.M., 2017. An Efficient Image Watermarking Method Based on Fast Discrete Cosine Transform Algorithm. *Mathematical Problems in Engineering*, 1–10. doi.org/10.1155/2017/3509258.

Verma V., Jha R.K., 2015. An Overview of Robust Digital Image Watermarking. *IETE Technical Review*, 1-18. doi:10.1080/02564602.2015.1042927.

Ye R., Zhuang L., 2012. Baker Map's Itinerary Based Image Scrambling Method and Its Watermarking Application in DWT Domain. *International Journal of Image, Graphics and Signal Processing*, 4(1), 12-20.

Yu X, Wang C, Zhou X., 2018. A Survey on Robust Video Watermarking Algorithms for Copyright Protection. *Applied Sciences*, 8(10), 1891. doi.org/10.3390/app8101891.

Zeki A.M., Manaf A.A., 2009. A Novel Digital Watermarking Technique Based on ISB (Intermediate Significant Bit). *World Academy of Science, Engineering and Technology*, 3, 444-451.

Zotin A., Favorskaya M., Proskurin A., Pakhirka A., 2020. Study of digital textual watermarking distortions under Internet attacks in high resolution videos. *Procedia Computer Science*, 176, 1633-1642.