International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

KBM Based Variable Size DCT Block Approaches for Video Steganography

Kemal Tutuncu[1,*], Murat Hacimurtazaoglu[2]

Submitted: 22/05/2021 **Accepted : 31/07/2021**

Abstract: In this study, the data hiding operations were performed by a combination of different sizes of Discrete Cosine Transform (DCT), Key Block Matrix (KBM) consisting of different sub-pattern blocks, and additional security algorithms. Mean Square Error (MSE), Structure Similarity Index Measurement (SSIM), and Peak Signal Noise Ratio (PSNR) quality metrics were used to evaluate the results of the proposed approach. 4x4, 8x8, and 16x16 pixel DCT blocks were used for testing and evaluation. Each DCT block was used with the proposed KBM that can be created by artificial intelligence techniques and secure data hiding methods. The fact that the KBM structures used in the proposed methods are variable increases data security in the proposed algorithms and fills the gap of security issues in the video steganography. The quality metric values obtained showed that the proposed approach can compete with the results obtained in the literature regarding PSNR, MSE, and SSIM values.

Keywords: *DCT, Variable DCT blocks, Secret Key, Blok Matris, PSNR, MSE, SSIM, Video Steganography*

1. Introduction

Data security is one of the biggest challenges of this era due to the wide usage of the internet and multimedia. Many studies have been carried out for the secure transmission or hiding of data. Hiding the data in multimedia tools is called steganography. Unlike cryptography, data in steganography is embedded in another multimedia without encryption. The multimedia tools such as music, video, text, or image files in which the secret data is hidden (embedded) are called stego-objects. Different methods and techniques are used in the studies related to steganography. Three parameters are taking into consideration whatever methods are used for implementing the steganography. These are the capacity of the hidden data, imperceptibility of the hidden data in stego-object, and robustness of stego-object against any attack. Robustness can be told as resistance against statistical attack, cutting and/or cropping the stego-object, or deleting some part of the stego-object.

Due to the human visual system is less sensitive to the small changes of digital media, especially for digital video, video steganography has become more famous recently. In this study, a new approach was proposed to improve the data security in video steganography. The proposed approach uses a KBM structure with a size of 64x64 pixels. KBM is not unique and can be changed in structure depending on user preference. For secret data hiding into the video frames, 3 different sizes of DCT block structures were used. These are 4x4, 8x8 and 16x16 pixel block structures. But the pixel in which the data is embedded is chosen with the combination of KBM, DCT blocks, and additional security algorithms.

By comparing the original image with the stego object obtained by each of the DCT block structures obtained by proposed methods, the image defects were examined. MSE, SSIM, and PSNR were used as quality metrics for comparison aim.

1.1. Video Steganography

Data hiding in video files generally use data hiding methods in pictures and audio files together. Moving images (digital videos) consist of still images and sounds that are sequenced in a row and play quickly. In this case, the data hiding technics used to hide data in picture or audio files are also used for data hiding in video files. Moving images are images obtained by sequentially replaying still digital images (pictures) 25 times per second or more. Here, each digital image is called a frame, and moving images are expressed using the number of frames per second (fps). Due to the inability of the human eye to be sensitive to frequencies above 25 Hz, still pictures are played 25 times or more per second.

Moving image recordings are recordings that offer much more data hiding compared to still image recordings. This increases the appeal of video recordings in data hiding. For example, a video file with a capacity of 40 fps and a length of 10 seconds consists of 400 pictures. Therefore, the data to be hidden in a 10-second video will be 400 times more than the data to be hidden in an image file. Also, the video image will continue to play without noticing possible errors or image defects that may arise from data hiding methods on each image that will create the video.

The quality of the video file depends on the number of frames it plays per second and the resolution of each frame. Each frame image sequence consists of rows and columns (m x n). Each element in rows and columns is called a pixel.

The colour in each frame that makes up the video file is a mixture

[1] *Department of Electric Electronics Engineering, Selcuk University Technology Faculty, Konya, 42130, Turkey*
ORCIDID : 0000-0002-3005-374X
[2] *Department of Computer Programming, Recep Tayyip Erdogan University, Ardesen Vocational School, Rize, 53400, Turkey*
ORCID ID: 0000-0001-6096-6526
* *Corresponding Author Email: ktutuncu@selcuk.edu.tr*

of red, green, and blue (RGB) colours. Each colour corresponds to 256 different colour tones expressed with an 8-bit numerical value between 0 and 255. Since each colour value is expressed with 8 bits, the pixel value representing each square of the video frame has a value of 24 bits.
Audio files can be defined as file types that contain sound waves at certain frequencies in their content. Hiding data in digital audio data is a much more difficult process than other media elements. It is a demanding process due to the frequency threshold limit of people's sound hearing. The wav file format, which provides uncompressed quality sound recording that takes a lot of memory, is among the most preferred audio recording formats. For this reason, it is often used as a cover file in which the data is hidden in audio steganography.

1.2. Literature Survey

In this section, literature studies using the DCT method and block-based methods will be presented. It will be focused on more the studies that are similar to the proposed study and based on DCT-based video steganography that makes use of the Least Significant (LSB) method.
In [1], data hiding was performed on DCT coefficients using the LSB technique. To prevent visual distortion in the study, data hiding was not applied to the values less than the threshold value or equal to 0 from the DCT coefficients obtained. Histogram analysis was used as the quality metric. As a result of the evaluation, it was stated that there was not much change between the original image and the carrier image.
In [2], LSB and DCT methods were applied on gif, bmp and jpg file formats and the results obtained were compared. It was stated that the PSNR values obtained from DCT-based steganography applications are higher compared to LSB-based steganography applications. The highest PSNR value obtained from steganography studies performed according to the DCT technique was found as 57.21 dB.
In [3], DCT and DWT based video steganography applications using the LSB method were performed. PSNR quality metric was used to evaluate the results of the study. It was stated that the highest PSNR value obtained from the applications was 29.80 dB. When studies that are partially similar to the DCT technique we used in our study and the method content we propose were evaluated.
In [4], the author performed data hiding in the image by using a method similar to the method of hiding data into DCT blocks that is proposed in this study. In the study [4], data hiding was performed on the compressed image using YCbCr colour space. The author suggested that data hiding on the Y value is safer. The results of the study were evaluated according to the PSNR quality metric. The highest PSNR value was obtained as 44.97 dB at hiding 40000 bits of data.
In [5], the pre-embedding model, which is one of the methods that is proposed in this study, was also examined. The performance evaluations of video steganography performed according to intra-embedding, pre-embedding and post-embedding techniques were conducted. It was stated that the highest PSNR value was obtained from the pre-embedding technique as 59.63 dB.
In [6], a steganography study was applied by dividing the image into blocks and encoding it. Fixed blocks of a 16-character size determined by the key were used. MSE and PSNR metrics were used to evaluate the application results. It's been seen that the best PSNR value obtained was close to 40 dB.
In [7], video steganography was performed using a macroblock application. The block structure initially consists of 16x16 pixel blocks. Afterwards, the block structure was divided into smaller blocks and data hiding was implemented. It was stated that the highest PSNR value obtained in the evaluation of the application results was 40.747.
Ref. In [8], data hiding was performed using Five Pixels Pair Differencing (FPPD) steganography by adjusting the pixel difference between 2x3 size blocks. Thus, the data was provided to be hidden in more than one cover image. It's been seen that the average PSNR value obtained in the study was 37.62 dB.
In the study conducted in Ref [9], data was hidden by tissue synthesis. The texture was reconstructed by synthesizing a new texture image with the texture in the image. The source texture image of the algorithm and the message to be hidden were embedded in the stego-image.
In the study conducted in Ref [10], video steganography was performed using DCT and DWT techniques based on the detection of scene changes. LSB method was used for data hiding. It was implemented by replacing the LSB of the 8x8 DCT block coefficient with message bits. It's been seen that the highest PSNR value obtained was close to 30 dB.
In the study conducted in [11], data was hiding according to the different pixel values using each colour channel (RGB). The highest PSNR value obtained in the study was 48.49 dB. In addition, histogram analysis was used for image quality.
In Ref. [12], a recess-based steganography study based on byte difference was performed in compressed video. Data hiding was implemented by using the coeficients of brightness components within 8x8 blocks. 8x8 blocks are mixed with the help of a specific key. Data embedding was conducted in the selected sub-blocks. The best PSNR resulting from the applications was around 46 dB.
The proposed method includes a 64x64 poly-pattern block matrix that can shift in four directions (up-down-right-left) and rotate up to 270^{o}, XOR and AND logical operations for choosing directions/degree and also next 64x64 predetermined block. Finally, the ability of choosing different channels of the RGB of a pixel to embed the data ensures the robustness of the proposed video steganography application. Experimental results have been showed that the proposed method has attained good perceptual quality compared to the other video steganography methods that includes classical DCT and other DCT based approaches.
The results obtained from the evaluation metrics show that the suggested methods work well. When the metric values obtained from the results of the previous studies are compared with the metric values obtained from our study, it is seen that the PSNR values we obtained are higher than the studies conducted.

2. Methodology

In this study, a block-based video steganography algorithm that makes use of the DCT method is proposed. Data hiding operations into the DCT blocks that have different structures are implemented according to KBM values. The pixels within the video frame where data can be hidden vary for each 64x64 pixel block according to the KBM pattern structure and DCT block structure.

2.1. DCT Block Definition

The DCT is similar to the Discrete Fourier Transform (DFT): it transforms a signal or image from the spatial domain to the frequency domain.
The DCT coefficient $I(u,v)$ of a video frame with MxN in size at $g(x,y)$ index is calculated according to equation (1) [3, 13].

$$T = \alpha_u \alpha_v \cos\frac{(2x+1)\pi u}{2M} \cos\frac{(2y+1)\pi v}{2N}$$
$$\alpha_u = \frac{1}{\sqrt{M}}, u = 0; \sqrt{2/M}, 0 < u \leq M-1;$$
$$\alpha_v = \frac{1}{\sqrt{N}}, v = 0; \sqrt{2/N}, 0 < v \leq N-1;$$

$$I(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x,y)T \quad (1)$$

The signal matrix obtained as a result of DCT applied to a video frame of MxN size is again formed in MxN size. The Inverse Discrete Cosine Transform (IDCT) operation is calculated according to equation (2) to restore the original image after the DCT operation.

$$g(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(u,v)T \quad (2)$$

Studies in the literature show that DCT is generally applied using blocks of 8x8 pixels. In this study, DCT and IDCT operations were performed using 4x4, 8x8 and 16x16 pixel blocks. The reason of using different DCT blocks is to compare the performances of different algorithms of the proposed approach.

2.2. Key Block Matrix (KBM)

The key block matrix was created with the aim of determining the data-hiding pixel areas and involves the binary-coded form of the key block structure. The key block structure consists of a 64x64 pixel main block and 16 pattern blocks in 16x16 pixel. The pattern blocks are randomly selected from 18 pre-coded source pattern blocks with a pixel size of 16x16. The selected pattern blocks are placed in the key block from left to right. Fig. 1 a shows the sample key block, Fig. 1 b shows the layout of the pattern blocks that constitute this block and Fig. 2 shows the pattern block matrix structure of pattern block number 5.

5	17	18	15
18	17	16	14
10	8	18	11
1	15	2	17

Fig. 1. a) Sample key block structure b) The layout of the pattern blocks that constitute the key block

After the creation of the key block, the KBM is created using 0 for pixel areas corresponding to the black colour (0) value in the block and 1 for pixel areas corresponding to the white colour (255) value. The pixel areas represented by 0 within the key block matrix are used as closed areas where data is not embedded while the pixel areas represented by 1 are used as open areas suitable/appropriate for data-hiding.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. The 16x16 pixel pattern block matrix structure of pattern block number 5

2.3. Performance Criterions

Video quality is a measure of the perceived video distortion of a video obtained by going through a particular video processing system compared to the original video [14].

Various metrics are used to approximate video quality. The most widely used metrics are PSNR and MSE. These metrics are based on a pixel-by-pixel comparison of source video signals and test video signals.

2.3.1. MSE

MSE is the mean of the squared difference between the corresponding pixel values of two images. It is an image quality evaluation based on the error sensitivity between the corresponding pixels. It gets an error value between 0 and 65025 for the 8-bit colour value [15]. This value is expected to be as small as possible for a high-quality video processing result. A value of 0 states that there is no difference between the two videos while 65025 represents the maximum difference for the 8-bit colour value. The closer the result is to 0, the more successful the video processing result is considered to be. The MSE value is calculated based on the following equation (3).

$$MSE = \frac{1}{n.m} \sum_{x=1}^{n} \sum_{y=1}^{m} (|f(x,y) - f'(x,y)|)^2 \quad (3)$$

In this equation (3), *m* and *n* represent image size, *f(x,y)* represents the pixel value for the original video image and *f '(x,y)* represents the pixel values for the test video image.

2.3.2. PSNR

PSNR is defined as the ratio between the maximum possible power of a signal and the noticeable power of the potential distortion (noise level in the video) in the processing and transmission of video [16]. The logarithmic decibel scale (dB) is generally used for PSNR due to the wide dynamic range of signals. If the calculated PSNR value is close to 1, it indicates that the quality of similarity between two images is high. The PSNR value is calculated based on the following equation (4).

$$PSNR = 10 \text{ x} \log_{10}\left(\frac{(f_x)^2}{MSE}\right) \quad (4)$$

This equation (4), f_x represents the value of maximum pixel or maximum signal in the image. It is calculated as $f_x = 2^n - 1$ for the maximum signal value with *n* representing the pixel bit value. For example, pixel values are calculated as $f_x = 255$ for an 8-bit colour value and as $f_x = 1023$ for 10-bit colour value.

2.4. Determining Pixel Areas for Hiding Data

Proposed steganography algorithms for data hiding are as follows:

- The KBM to be used will be selected by the user from one of the fixed, shifting, rotating and shifting-rotating structures,
- With the key block matrix previously formed, data hiding is performed using one of the Sequential, Sequential in All Frames or Non-Sequential in All Frames methods based on preference,
- Hiding data in the RGB, R, G or B colour channel depending on user selection,

Thus, the user has the following options to choose fields (pixels) in the frames to hide data. Fig. 3 shows all the options or algorithm combinations to hide data.

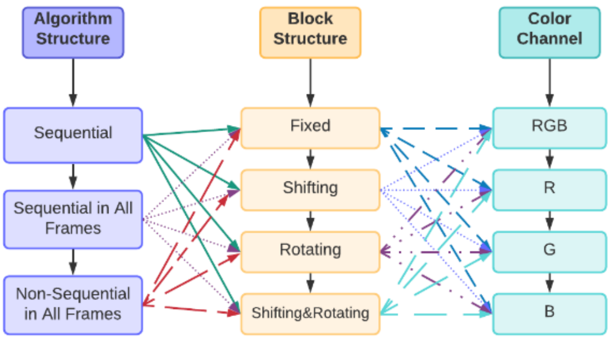


Fig. 3. Algorithm selection for data hiding

In algorithms using the fixed block structure, KBM generated is used fixedly for each DCT block located on the 64x64 pixel area on the video frame. If the KBM fields corresponding to the first bits of the DCT blocks are suitable for hiding data, data is hidden within the selected colour channel. The processes continue using a fixed 64x64 KBM from left to right for each 64x64 pixel block area until the entirety of the data to be hidden is embedded within the video frames.

In data hiding algorithms to be made using shifting, rotating and shifting-rotating block structures, KBM is placed in the first 64x64 pixel area of the video frame that will be used for data hiding. After data hiding according to DCT block structures in this area, the KBM structure changes for the next 64x64 pixel block area. Changes in the KBM structure are made according to the colour channel selection made and the first pixel value of the DCT block area where the last data hiding process was performed.

KBM can rotate 90^o, 180^o and 270^o in right-left and up-down directions. The movement direction and step controls of DCT blocks and KBM are selected according to the DC coefficient of the DCT block structure. In motion controls, each step is expressed as a pixel.

The steps to determine the next KBM structure are as follows.

Step 1. The DC value of the latest DCT block that is used to hide data within the 64x64 pixel block is read.

Step 2. For the selected DC value and RGB colour channel,

ch1 = DC value for B colour channel,

ch2 = DC value for G colour channel,

ch1 is used for motion control for right–left direction,

ch2 is used for motion control in the up-and-down direction,

Here, the DC value for each colour channel is represented in the binary system and 6th bit means control bit and 5., 4., 3. and 2^{nd} bits represents the number of steps in pixel.

As an example, let's assume that B = 215, G = 187 and, R = 96 in data hiding operation that will use RGB colour channel;

ch1 = B = 11010111

ch2 = G = 10111011

For ch1 colour value,

7	6	5	4	3	2	1	0
1	1	0	1	0	1	1	1

215 = 11 0101 1 1

Control bit (1^{th} bit and expressed as ch1_control) = 1

The number of steps (ch1_step) = $(0101)_2$ = 5 in pixel.

For ch2 colour value,

7	6	5	4	3	2	1	0
1	0	1	1	1	0	1	1

187 = 10 1110 1 1

Control bit (1^{th} bit and expressed as ch2_control) = 1

The number of steps (ch2_step) = $(1110)_2$ is calculated as = 14 pixel.

Step 3. The KBM block structure that will be used for the next 64x64 pixel block will be created according to the calculated ch1 and ch2 values.

According to these calculated values;

- For (ch1_control = 0) or (ch1_control = 1),

  0 = Shifting right process

  1 = Shifting left process

  ch1_step = represents number of steps for shifting
- For (ch2_control = 0) or (ch2_control = 1),

  0 = Shifting up process

  1 = Shifting down process

  ch2_step = represents number of steps for shifting

KBM is shifted by the number of pixels in the direction obtained by control value. For the operation sequence, first the right-left shifting process and then the up-down shifting process is applied. For our example, shifting left in 5 pixels and shifting down in 14 pixels will form new KBM structure.

- For (ch1_control & ch2_control == 0) or (ch1_control & ch2_control == 1),

  0 = Rotating right process

  1 = Rotating left process

  ch1_step = Number of steps to rotate right

  ch2_step = Number of steps to rotate left.

When the steps are considered (ch1_step, ch2_step);

1: KBM will return 90^o in the direction of the rotation

2: KBM will return 180^o in the direction of the rotation

3: KBM will return 270^o in the direction of the rotation

4: It is stated that the KBM will remain constant.

Since the key block matrix will be equal to itself after every 4-step rotation, the shifting process is carried out in the direction determined with the control value that is equivalent to the remainder of the division of the number of steps to 4.

step = ch1_step % 4 or

step = ch2_step % 4

If the user chooses shifting and rotating operation for KBM structure then first shifting, and then rotating operations are performed on the KBM to have a new KBM.

According to the structure of the sequential data hiding algorithm, the data is hidden in DCT blocks in a left-to-right order according to the KBM value. If the KBM value corresponding to the first bit of the DCT block is suitable for hiding data, data embedding is performed within the DCT block. The direction and status of the next DCT block is determined by moving the data on the area determined by the 64x64 pixel KBM pattern structure of the first bit data of the hidden DCT block.

In the selection of sequential data hiding algorithm within the entire video (within all frames), the pixel areas suitable for data hiding on all video frames that make up the video are determined by marking the DCT blocks and the pixel areas of these blocks that are suitable for data hiding on the KBM. The amount of data to be hidden in each video frame is determined by taking the ratio of the total length of data to be hidden to the total video frame length. The data bits are sequentially distributed in DCT blocks suitable for data hiding in the frame.

In a non-sequential data-hiding algorithm, the only difference is embedding the data in all frames non-sequentially according to DCT block and KBM. The determination of the pixel areas where data will be hidden is calculated as in the process of sequential data-hiding in all frames. The common point for embedding data in all frames algorithms is to embed an equal amount of data into

each frame.
DCT blocks are 4x4 pixel, 8x8 pixel and 16x16 pixel. In line with the algorithm selected, data hiding operations are performed according to each DCT block structure.
The process steps are as follows to determine the pixel areas suitable for hiding data according to the algorithm selections that the user will make.
Step 1. The KBM to use in the data hiding process is determined.
Step 2. The 64x64 pixel area to be processed on the video frame is divided into the size of the selected DCT block.
For example, the DCT block is selected in 8x8 pixel size. In this case, the areas to be processed on an area of 64x64 pixel will be as shown in Fig. 4.

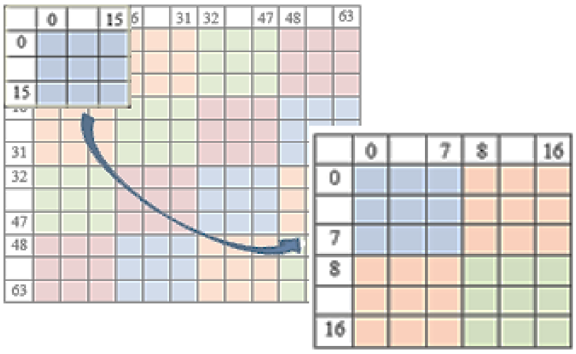


Fig. 4. 8x8 DCT block representation on 64x64 pixel area

There are 4 DCT blocks within each 16x16 pixel area for an 8x8 pixel DCT block. In this case, if direction and status values are not considered within each 64x64 pixel area, a minimum of 64 bits of data can be hidden for each colour channel.
Assume that a sample 64x64 pixel KBM whose first 16x16 pixel block area is as in Fig. 5 is located on the 1st 64x64 pixel block are of the video frame.

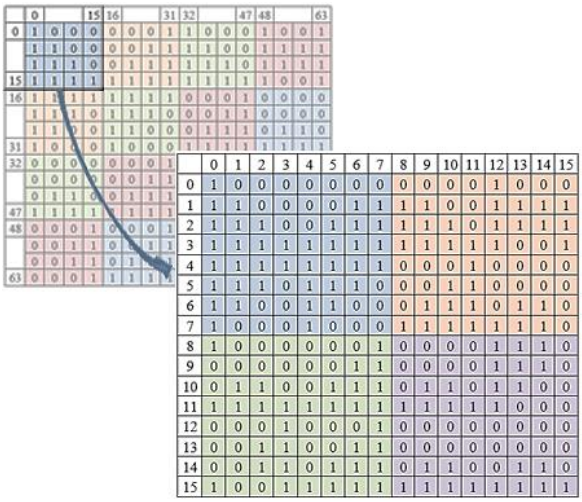


Fig. 5. 16x16 pixel block area of KBM

If and only if an 8x8 DCT block is used for data embedding 4 bits of data can be hidden into the 16x16 pixel area in the frame. When KBM in Fig.5 is used with 8x8 DCT block, it will be seen that;

1. The KBM value (0x0) corresponding to the 1st bit of the 8x8 DCT block is 1 and data can be hidden in the DCT block,

2. KBM value (0x8) corresponding to the 1st bit of the 8x8 DCT block is 0 and data cannot be hidden in the DCT block,

3. The KBM value (8x0) corresponding to the 1st bit value of the 8x8 DCT block is 1 and data can be hidden in the DCT block,

4. KBM value (8x8) corresponding to the 1st bit value of the 8x8 DCT block is 0 and data cannot be hidden in the DCT block.

In this case, it's seen that 2 bits of data can be hidden within a block area of 16x16 pixels without checking the status & direction.
If state&direction controls are considered for KBM in Fig. 5, the number of the pixel that the data can be embedded will be at least 1 bit and at most 2 bits. According to the KBM structure, the operations for the next 64x64 block structure are continued by location and directional control.

2.5. Intra-block Motions & Hiding Data

In the process of determining the pixel areas where data can be hidden in the video, data is hidden in each pixel area suitable for data hiding according to the selection of data hiding algorithms and the KBM structure.
Each video frame is divided into 64x64 pixel blocks to represent an integer. DCT blocks are placed in these reserved pixel areas according to the DCT block selection. The last 64x64 pixel block of each video frame is separated into the control block used to hide selected in the data hiding.
In this case, using 8x8 DCT blocks for each frame of a video of 640x480 p, the number of blocks that can be data hidden is calculated as follows.

Horizontal blocks = int (640/64)

Number of horizontal blocks = 10

Vertical blocks = int (480/64)

Number of vertical blocks = 7

Calculation results show that number of available 64x64 pixel blocks are 69 for data hiding within each video frame, with the last 64x64 pixel area reserved for hiding control values.
For example, for an 8x8 pixel DCT block, 64 DCT blocks in each 64x64 pixel block area can be used for data hiding. 4416 (64x69 = 4416) DCT blocks can be used for 69 block areas (64x64 pixel) in the video frame. The motion process steps between DCT blocks within each 64x64 pixel block area are as follows.
Step 1. A block area of 64x64 pixels is parsed into blocks with a structure of 8x8 pixels.
Step 2. As shown in Fig. 4 and Fig. 5, the KBM corresponding to the first bit of the first 8x8 pixel block area is checked. If the DCT block area is suitable for hiding data, the field is processed to hide the data.
Step 3. In the area determined according to the DCT block structure, DCT conversion is performed for each colour channel to be used in the data hiding process and the DC coefficient of each DCT block is calculated.
Step 4. According to the calculated DC coefficient value, the direction and status control of the next DCT block to be moved is performed.
Step 5. Direction and step counts are determined for the shifting and rotating movements on the existing KBM to form the next KBM structure according to the DC coefficient value of the most recently processed DCT block for data hiding.
According to the selection of the data hiding algorithm made after determining the pixel areas suitable for data hiding in the video frame, the data to be hidden in the frame is hidden in the appropriate areas marked.
The data is hidden by using the fields (3,0) and (2,1) of the matrix block obtained as a result of the calculation of the DCT block coefficients. As an example, the cells to be used for data hiding for 8x8 DCT blocks are shown in Fig. 6.

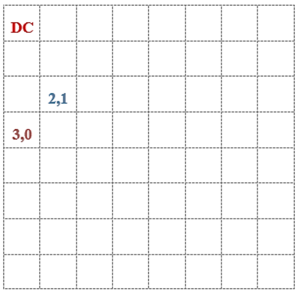


Fig. 6. Cells used for data hiding

The matrix values according to the hidden data value are as in equality (5).

$$\begin{cases} dct[3,0] \geq dct[2,1], data = 1 \\ dct[3,0] < dct[2,1], data = 0 \end{cases} \quad (5)$$

If the data to hide is 1, cell (3.0) is expected to be bigger than the cell (2.1). If equality is not achieved, cell (3.0) and cell (2.1) are replaced. Cell (3.0) is increased 1 if the cells are equal. If the data to hide is 0, cell (3.0) is expected to be smaller than the cell (2.1). If equality is not achieved, cell (3.0) and cell (2.1) are replaced. Cell (3.0) is decreased 1 if the cells are equal. No changes are made to the DC coefficient in the data hiding process.

3. Proposed Method

The proposed video steganography structure is shown in Fig.7. It mainly consists of the processes of creating the stego object (encoding) and obtaining the hidden message (decoding). Data hiding /encoding and data extraction/decoding are carried out in accordance with the KBM structure and DCT blocks mentioned previously.

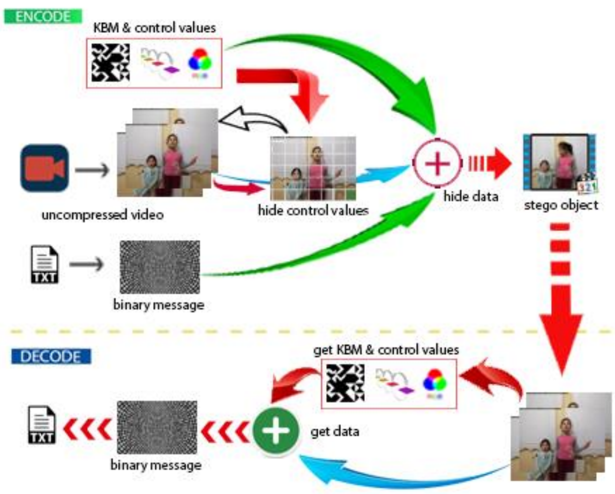


Fig. 7. Proposed Video Steganography Structure

A video has been recorded to be used as a stego object in the encoding process. The video size and the number of frames that make up the video are calculated to determine the video frames to hide data. As a result of user selection, the algorithm and KBM structure to be used in data hiding are placed in the last 64x64 pixel block areas of the video frames as the control values. The message is hidden in the video with the selected algorithm and KBM structure.

For decoding, the stego object is split into video frames. The KBM structure and algorithm used in data hiding are extracted by reading the control values from the last 64x64 pixel block area of the video frames. Using the extracted algorithm and KBM, the secret message hidden in the video is read.

3.1. Encoding Process

The methods used to generate the stego object are created by the user making a selection from 3 different sections from Fig. 3. First, the user selects the algorithm to be used in the data-hiding process among algorithms designed as sequential, sequential in all frames and non-sequential in all frames. After the selection of the algorithm, the fixed, rotating, or shifting-rotating KBM to be used in the data-hiding process is selected. Afterwards, the colour channel where data will be hidden is selected as RGB, R, G or B.

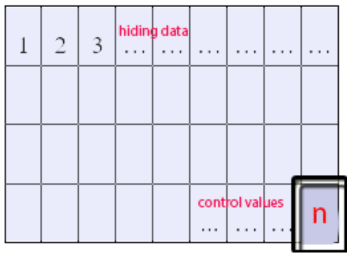
The user gives a name to the .txt file including the data to be hidden. The ASCII data input by the user in the txt format is converted into the corresponding values in the binary system and binary input data is obtained.

Video recording is made for the cover object to be used in the creation of the stego object. The dynamically produced cover object is created through the user recording a momentary image with a camera. Video frames and audio files are recorded separately to be combined afterwards. Each video frame is given a number based on the order of recording in the folder and saved as .png while the audio file is saved as .wav.

KBM, whose structural features were explained in detail previously, is used to be implemented in data-hiding. In accordance with the KBM selection, the structure of KBM changes or remains fixed during the movement on the 64x64 pixel blocks within each video frame. Within each 64x64 pixel block, the data is hidden in the selected DCT block matrix according to the KBM structure.

3.1.1. Embedding Control Value

In order to extract hidden data, the method parameters to be applied to create the stego object, the data size to be hidden, the colour channel, the DCT block structure and the KBM structure are recorded as control values. Control values are hidden in the final 64x64 pixel area of each video frame where data can be hidden.



Hiding the control values is performed in the following frames sequentially, starting from the first video frame where data can be hidden. After each control value is hidden sequentially, one video frame is left blank and the hiding from the next video frame is continued.

Control values other than the KBM control value are hidden in a similar structure within the pixel area where they will be hidden. However, each pattern forming the KBM structure may differ structurally. In this case, since the receiving party does not have any information about the patterns and their sequences, the pattern structure that makes up the KBM is recorded as a whole. The following shows the process of hiding the KBM control value.

Hiding the KBM control value:

A layout matrix is created in order to hide KBM values inside the

video frames. The layout matrix is created in the width of the pattern block matrix forming KBM and in size to accommodate each pattern block (layout matrix size [16x16,16] as in Fig. 1).
The first-row elements of the created layout matrix are created by taking the first elements of each pattern block matrix in order from left to right. Operation continues for each element of the following row of the layout matrix by taking the following element of the pattern block matrix. In this case, each row element of the pattern block matrices is placed vertically in the layout matrix. In order to prevent this situation, during the placement of the layout matrix in the video frames, each pattern is shifted to the right by the index value in the row layout matrix.
The algorithm for hiding the key block matrix in the video frame is as follows:

Step 1. A layout matrix is created in which the values of each 16x16 pixel pattern block matrix forming a 64x64 key block matrix are recorded, starting from the first element, in order of each pattern block matrix elements.

Let the key block matrix be as follows:

1010	0011	1100	1010
1001	1010	1111	0110
1110	1010	1100	0111
1110	1001	1101	1111

The first line elements of the layout matrix consist of areas of pattern block matrices marked with red colour.
1011111011101111
The second line elements of the layout matrix consist of areas of pattern block matrices marked with blue colour.
0010001110111011
The third line elements of the layout matrix consist of areas of pattern block matrices marked with green colour.
1101011111011001
All elements of pattern block matrices up to 16x16 matrix size are recorded in the layout matrix in order.
1011111011101111
0010001110111011
1101011111011001
0100101000010111
…

Step 2. The pattern block matrix values written in the layout matrix are shifted to the right by the index value they are in the layout matrix.
0. Indians - 101111011101111
1. Indians - 0010001110111011
2. Indians - 110101111011001
3. Indians - 0100101000010111

The values that make up the 1st row of the layout matrix are kept constant according to the index values (0) they are placed in the matrix (0 steps are turned to the right).
The values that make up the 2nd row of the layout matrix are rotated 1 pixel right according to the index values (1) they are placed in the matrix.
The values that make up the 3rd row of the layout matrix are rotated 2 pixels to the right according to the index values (2) they are placed in the matrix. A placement matrix is formed by rotating the row elements of each pattern block consisting of 16 rows to the right according to the 0 - 15 index value they are placed in the placement matrix.
The first element of each pattern block matrix constitutes the first element of the layout matrix, and the last elements of the first row of each pattern block matrix constitute the 16th row element of the layout matrix. These 16 lines created are considered as 16x16 blocks. The new block, which will start by taking the second-row elements of each pattern block matrix, creates the second 16x16 block, and each subsequent new row creates a new 16x16 block. At the end of 16 rows, [16x16,16] layout matrix is created. Unnecessary repetition of operations will occur in the shifting operation, which will be performed as an index value according to the 0 - 16x16 index value for the shifting of the data in each row of the layout matrix. In order to prevent these repetitions, 0 - 15 pixel shifting is performed for each 16x16 block (index mode 16). The aim here is to prevent the pattern block matrix elements from being placed vertically or horizontally in the same column to increase security.

Step 3. The control data is placed in the layout matrix in the data hiding pixel areas that are reserved for hiding the control data starting from the 1st video frame.
The DC value of DCT blocks in the 64x64 pixel block area reserved for hiding the control values on the video frames is as follows.

7	6	5	4	3	2	1	0

If XOR($R_{(7-1)}$ AND XOR($G_{(7-1)}$ AND XOR ($B_{(7-1)}$)) = = 0 the first element of the layout matrix is hidden in the DCT block here. It moves sequentially from left to right within a 64x64 pixel block. To determine next pixel area the logical expression of XOR($R_{(7-1)}$) AND XOR($G_{(7-1)}$) AND XOR ($B_{(7-1)}$) = = 0 is checked. If the result is 0, data is hidden within the pixel area. If the result of the operation is 1, data is not hidden in this pixel area. The process is continued until all elements of the layout matrix are embedded.
For example, hiding the colour channel value from control parameters with a similar structure is as follows.

Hiding the colour channel control value:
The block area reserved for hiding the control values is checked for data hiding in order from left to right, starting from the first pixel. The colour channel information selected for data hiding to the R, G and B colour channels of the pixel area suitable for data hiding is hidden in the DCT block.

3.1.2. Embedding Data

In line with the selected algorithm, the amount of data to be hidden in each video frame is calculated, taking into account the capacity of the video frame and the length of the message. The data is hidden in the video frame, the last 64x64 pixel block area to be allocated to the control data, starting from the first 64x64 pixel block until the last 64x64 block matrix structure sequentially according to the DCT block structure and KBM structure. Thus, data hiding is performed with the user selected algorithm. After the data is hidden in each video frame in accordance with the selected algorithm, a new video frame called the stego object is created. When the data hiding process is completed, the resulting video frames and audio files are combined to create a stego object in the '.avi' file format.
For the process of hiding sample data;
- Let the video that will be used for data embedding be dynamically recorded (339 frames),
- Let secret text data entered by the user has 18.4 kb size (93369 bits in a binary system),
- DCT block structure is 8x8 pixel size as default,
- Let the *"RGB"* colour channel, shifiting&rotating KBM block structure and Sequential Data Hiding in All Frames algorithm are chosen by the user.

According to these values, steps of the data hiding process are as follows:

Step 1. A 64x64 pixel KBM (Fig. 8.a)) is created by randomly selected 16 pattern blocks (Fig. 8.b) among previously created 18 pattern blocks.

2	8	13	8
15	17	14	4
8	14	4	15
7	8	6	4

Fig. 8. a) Placement of kmb b) pattern blocks produced

Step 2. The size of the data to be hidden in each video frame is calculated. For this example, 275 bits of data are hidden in each video frame.

Step 3. In order to extract the hidden message in the decoding process, the control parameters are hidden in the last 64x64 pixel area of the video frames.

Step 4. When the sequential data hiding in all frames, RGB colour channel and shifting-rotating KBM structure are considered;

-Each video frame is divided into 64x64 pixel blocks, sequentially,

-Each 64x64 pixel block is divided into 8x8 pixel blocks for the DCT block structure,

-Data hiding is performed according to the selected colour channel, KBM structure and the suitability for data hiding for the pixel area that the movement index represents.

According to the colour channel selection and appropriateness for data hiding regarding the first bit of the segmented DCT block matrix of the video frame corresponding to the KBM data-hiding area, the first three data bits are hidden in the DCT block. After the first data block is hidden, a state & direction check is made for the transition to the next DCT and 64x64 pixel blocks. The next step is to check whether the data can be hidden or not in the destination pixel. After this step, in order to hide data for the next DCT block, the corresponding field in the KBM structure must be suitable for data hiding and data hiding state control data must be suitable for data hiding.

The KBM structure to be used for the next 64x64 pixel block is changed according to the DC coefficient value of the 8x8 DCT block, which is the last data hidden in the first 64x64 pixel block structure.

Briefly speaking; According to the DCT block structure, video frames are divided into blocks. KBM is positioned on the video frame. All the channels of RGB are currently selected as the colour channels to hide data. The suitability of data hiding of the corresponding area on KBM of the first pixel of the DCT block is checked. According to the selected colour channel information, 3 bits of data are hidden in the first DCT block in KBM in which data can be hidden.

Step 5. After the data hiding process is completed ". avi" format stego object is created.

3.2. Decoding Process

In the process of obtaining the secret message, the receiving party does not have the information about control parameters that are used to hide the message (KBM structure, colour channel, algorithm and DCT block structure). The user only uses the stego object as the input parameter. With which control values the message is hidden, the data is also obtained from the stego object. Basically, the structure of the decode process is similar to that of the encode process. Control parameters selected by the user in the encoding process are extracted from the stego object in the decoding process.

For the data extraction process, the user is primarily asked to identify the stego object, which is the input parameter. In the next step, control parameters are obtained from the stego object by reading the data hidden in the last 64x64 pixel block areas of video frames. Determining the pixel areas (suitable fields) where control data may be hidden is in the same structure as determining the pixel areas suitable for data hiding in the process of hiding control values. Considering the block area structure where the control data is placed in the video frames and the situation of passing a video frame blank as a result of the placement of each control parameter, possible video frames in which the control values may be hidden are determined. Control data are obtained in the order of embedding.

According to the KBM block structure, algorithm structure, colour channel used, DCT block structure, and the selected algorithm obtained from the control parameters, the data hidden in the stego object are extracted and encoded in binary format. Finally, a secret message is obtained by transforming the data obtained in binary format into ASCII characters.

4. Application Results

In this section, the results of the assessment of the proposed algorithms were presented. For data-hiding, the user was initially asked to input the file path where the message was located and record the video where data will be hidden. After the selection and video recording, the user selects the block structure, data-hiding algorithm, and colour channel where data will be hidden with the aim of enabling the receiver to be able to read the data.

Applications were implemented separately for DCT blocks of 4x4, 8x8 and 16x16 pixels.

The Python PyCharm IDE was used in the program-writing process for data-hiding and outcome analysis.

To evaluate the proposed methods, a dynamically recorded video was used. Video frames and audio files belonging to this video were used for each application. The video has a pixel resolution of 640x480 pixels and 330 video frames.

A 64x64 pixel KBM consisting of the same pattern blocks was used for each application to ensure application reliability. In the progressing steps of each application, KBM may vary depending on the selected block structure.

Data hiding operations are performed according to the selected algorithm structure and DCT block structure.

The comparison of the original and stego frames was made based on the PSNR, MSE and SSIM values. The results obtained are displayed comparatively for each application in tables.

The message data capacities that are hidden and obtained for each recommended application are shown in Table 1.

In sequential data-hiding algorithms, quality assessment is performed by considering the number of data-hidden blocks. Otherwise, including the frames with no data in the assessment will reduce the reality of the result obtained. For a healthy assessment of the results, the frames used by each algorithm in data-hiding were used in the assessment process. Frames with no data were excluded from the assessment. The quality metrics obtained from the results of sequential data hiding operations performed without using KBM according to the 4x4, 8x8 and 16x16 DCT block structure of the 1^{st} secret message into the stego object are presented in Table 2 whereas the results of data hiding operations using proposed data hiding algorithms and KBM are shown in Table 3, Table 4, and Table 5.

Table 1. Secret Message Data

		Input	Output
1st Secret Message	Message Length	93369 bit	93369 bit
	Message Size	10.4 kb	10.4 kb
2nd Secret Message	Message Length	376193 bit	376193 bit
	Message Size	42.8 kb	42.8 kb

Table 2. Quality Metrics Results of DCT Method for 1st Secret Message

Data Hidden Colour Channel	4x4 DCT Blok			8x8 DCT Blok			16x16 DCT Blok		
	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
RGB	15,92254	0,99674	36,11867	17,87768	0,99304	35,63546	17,58913	0,99210	35,71097
R	5,78320	0,99723	40,51432	7,65424	0,99404	39,31562	8,86108	0,99214	38,68979
G	7,63262	0,99691	39,31153	10,66271	0,99343	37,86015	13,48233	0,99165	36,85898
B	8,22243	0,99674	38,98708	11,09824	0,99346	37,69555	18,92805	0,99122	35,39671

Table 3. Quality Metrics Results of proposed data hiding algorithms and KBM for 1st Secret Message 4x4

Block Structure	Data Hidden Colour Channel	Sequential Data Hiding			Sequential Data Hiding in All Frames			Non-Sequential Data Hiding in All Frames		
		MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
Fixed	RGB	1,50670	0,99965	46,64887	0,27115	0,99990	54,75874	0,31471	0,99991	55,43243
Fixed	R	0,54028	0,99968	50,97895	0,09212	0,99991	59,13129	0,11032	0,99992	58,44801
Fixed	G	0,66951	0,99969	50,04672	0,11980	0,99992	59,38843	0,15929	0,99992	58,18687
Fixed	B	0,76449	0,99968	49,66807	0,25992	0,99989	55,35304	0,10404	0,99992	59,25749
Shifting	RGB	1,52626	0,99967	46,44525	0,36845	0,99992	56,19064	0,28458	0,99993	56,19101
Shifting	R	0,58481	0,99969	50,58130	0,13959	0,99991	59,14419	0,11508	0,99992	59,07317
Shifting	G	0,94886	0,99963	48,67727	0,14922	0,99991	57,77217	0,13281	0,99992	58,56042
Shifting	B	0,83823	0,99963	49,11223	0,09154	0,99991	59,98583	0,10131	0,99991	60,04656
Rotating	RGB	1,58137	0,99963	46,95130	0,16654	0,99993	58,18055	0,21438	0,99993	58,42683
Rotating	R	0,57887	0,99969	51,26133	0,09039	0,99991	59,70859	0,08351	0,99992	60,24681
Rotating	G	0,85311	0,99965	48,89081	0,21912	0,99990	56,53304	0,16174	0,99991	56,38052
Rotating	B	1,09713	0,99958	47,95676	0,17856	0,99989	56,13972	0,18310	0,99989	56,75181
Shifting-Rotating	RGB	1,74015	0,99966	45,81384	0,46859	0,99992	52,76195	0,31027	0,99991	55,49237
Shifting-Rotating	R	0,63708	0,99968	50,14305	0,18707	0,99991	55,93208	0,12862	0,99991	58,27569
Shifting-Rotating	G	1,01638	0,99963	48,20389	0,12777	0,99992	60,86346	0,17018	0,99990	58,29410
Shifting-Rotating	B	0,82184	0,99961	49,30808	0,15925	0,99990	56,65873	0,17641	0,99990	56,82489

Table 4. Quality Metrics Results of proposed data hiding algorithms and KBM for 1st Secret Message 8x8

Block Structure	Data Hidden Colour Channel	Sequential Data Hiding			Sequential Data Hiding in All Frames			Non-Sequential Data Hiding in All Frames		
		MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
Fixed	RGB	4,21581	0,99875	42,01812	0,91775	0,99944	48,69485	0,91932	0,99950	48,68003
Fixed	R	1,67769	0,99858	45,96107	0,47729	0,99917	51,42760	0,41528	0,99920	51,95814
Fixed	G	2,38041	0,99848	44,42347	0,84654	0,99921	49,51818	1,16405	0,99909	48,08577
Fixed	B	2,86344	0,99831	43,95962	1,34414	0,99908	47,12631	1,14300	0,99915	47,80443
Shifting	RGB	4,60665	0,99881	41,68043	1,00319	0,99957	48,74301	1,02623	0,99957	48,59588
Shifting	R	1,92298	0,99863	45,39682	0,46330	0,99928	51,61130	0,50950	0,99925	51,14011
Shifting	G	2,21658	0,99867	44,81133	0,82442	0,99923	49,67027	0,75829	0,99926	49,88559
Shifting	B	2,91194	0,99840	43,95659	1,23905	0,99916	48,20148	1,77149	0,99908	46,90163
Rotating	RGB	3,40337	0,99879	43,02417	0,88565	0,99949	48,80190	0,75125	0,99963	49,55425
Rotating	R	1,55326	0,99863	46,47500	0,48630	0,99916	51,39672	0,40815	0,99923	52,04492
Rotating	G	2,49347	0,99854	44,28147	0,89065	0,99925	49,37017	1,32389	0,99912	47,90238
Rotating	B	2,83298	0,99846	43,77303	0,90057	0,99913	48,76565	1,12360	0,99906	47,97516
Shifting-Rotating	RGB	4,58225	0,99875	41,75928	1,07919	0,99956	48,29249	0,92773	0,99958	49,04306
Shifting-Rotating	R	1,99385	0,99863	45,29010	0,51468	0,99919	51,15488	0,48537	0,99923	51,42779
Shifting-Rotating	G	2,28496	0,99862	44,65540	0,93093	0,99919	48,90599	0,80991	0,99918	49,46168
Shifting-Rotating	B	2,94444	0,99848	44,07274	1,06351	0,99927	48,39121	1,40325	0,99905	46,80210

Table 5. Quality Metrics Results of proposed data hiding algorithms and KBM for 1st Secret Message 16x16

Block Structure	Data Hidden Colour Channel	Sequential Data Hiding			Sequential Data Hiding in All Frames			Non-Sequential Data Hiding in All Frames		
		MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
Fixed	RGB	4,99276	0,99845	41,36309	4,41354	0,99828	42,18440	0,10958	0,99983	59,58088
Fixed	R	3,05606	0,99673	43,37883	2,81668	0,99672	43,84312	0,10466	0,99961	58,02030
Fixed	G	5,05947	0,99661	41,19578	5,27418	0,99643	41,09977	0,32292	0,99959	54,08264
Fixed	B	5,33029	0,99673	41,26433	5,29204	0,99696	41,22451	0,63023	0,99954	51,57041
Shifting	RGB	3,51907	0,99884	43,31786	3,87617	0,99887	42,85678	0,08093	0,99991	62,14237
Shifting	R	1,62168	0,99830	46,21026	1,49558	0,99820	46,71873	0,04970	0,99980	61,19982
Shifting	G	3,69322	0,99796	42,55825	4,49700	0,99798	41,71193	0,25976	0,99973	55,87902
Shifting	B	5,03851	0,99759	41,53216	4,78714	0,99775	41,60974	0,36580	0,99971	54,37388
Rotating	RGB	3,54794	0,99878	43,12411	3,08007	0,99889	44,16945	0,09181	0,99988	61,68739
Rotating	R	2,04103	0,99766	45,21541	1,94784	0,99754	45,47199	0,07534	0,99973	59,73029
Rotating	G	3,13318	0,99741	43,24948	3,68029	0,99719	42,60658	0,19929	0,99969	56,30112
Rotating	B	4,56455	0,99731	41,57283	4,59678	0,99745	41,57179	0,51274	0,99965	51,70026
Shifting-Rotating	RGB	2,05847	0,99916	46,37967	2,25917	0,99909	46,15733	0,04773	0,99993	64,50941
Shifting-Rotating	R	1,51333	0,99798	46,53187	1,75922	0,99789	45,75418	0,05956	0,99975	60,61908
Shifting-Rotating	G	3,14551	0,99823	43,86221	2,37580	0,99849	45,07032	0,22134	0,99976	56,53749
Shifting-Rotating	B	5,23983	0,99797	41,70631	5,85948	0,99786	40,73101	0,42605	0,99971	54,25787

Table 6. Quality Metrics Results of DCT Method for 2nd Secret Message

Data Hidden Colour Channel	4x4 DCT Blok			8x8 DCT Blok			16x16 DCT Blok		
	MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
RGB	15,61705	0,99673	36,20693	17,74427	0,99372	35,65176	17,26261	0,99248	35,88996
R	5,63701	0,99723	40,63297	7,94410	0,99388	39,15449	8,91848	0,99256	38,72403
G	7,52530	0,99693	39,39026	11,50444	0,99326	37,53676	12,41855	0,99235	37,22640
B	7,81948	0,99683	39,22795	12,41353	0,99321	37,26081	15,39894	0,99210	36,27345

Table 7. Quality Metrics Results of proposed data hiding algorithms and KBM for 2nd Secret Message 4x4

Block Structure	Data Hidden Colour Channel	Sequential Data Hiding			Sequential Data Hiding in All Frames			Non-Sequential Data Hiding in All Frames		
		MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
Fixed	RGB	1,24261	0,99966	47,27198	0,17882	0,99988	58,38310	0,16342	0,99988	58,25423
Fixed	R	0,46661	0,99969	51,56205	0,09249	0,99982	58,88284	0,10146	0,99981	58,35528
Fixed	G	0,60793	0,99968	50,57370	0,18015	0,99983	55,85579	0,21769	0,99982	55,57736
Fixed	B	0,78321	0,99964	49,22701	0,23711	0,99981	55,39600	0,33662	0,99979	53,28782
Shifting	RGB	1,54074	0,99967	46,59648	0,06851	0,99990	61,17168	0,06834	0,99990	61,34420
Shifting	R	0,57372	0,99970	50,77840	0,07639	0,99983	59,65721	0,07406	0,99983	59,88933
Shifting	G	0,88184	0,99965	48,79285	0,35852	0,99977	53,13025	0,23503	0,99980	54,55279
Shifting	B	0,99103	0,99962	48,25418	0,16779	0,99981	57,56936	0,20823	0,99980	56,47803
Rotating	RGB	1,24348	0,99966	47,59022	0,19440	0,99988	57,20666	0,18210	0,99988	58,19771
Rotating	R	0,44702	0,99970	51,98161	0,08995	0,99983	58,90884	0,09607	0,99983	58,57231
Rotating	G	0,66666	0,99966	49,98118	0,30002	0,99980	53,82852	0,35700	0,99978	52,80382
Rotating	B	0,99382	0,99960	48,23085	0,26602	0,99978	54,18208	0,20044	0,99980	55,35290
Shifting-Rotating	RGB	1,52876	0,99964	46,52543	0,12315	0,99989	60,13771	0,08540	0,99989	60,84801
Shifting-Rotating	R	0,55921	0,99969	50,83129	0,07787	0,99983	59,65810	0,07460	0,99984	59,76895
Shifting-Rotating	G	0,94473	0,99964	48,54225	0,34275	0,99977	54,74787	0,32528	0,99979	54,81098
Shifting-Rotating	B	0,87452	0,99961	48,93651	0,11549	0,99982	59,06476	0,15922	0,99980	56,71162

Table 8. Quality Metrics Results of proposed data hiding algorithms and KBM for 2nd Secret Message 8x8

Block Structure	Data Hidden Colour Channel	Sequential Data Hiding			Sequential Data Hiding in All Frames			Non-Sequential Data Hiding in All Frames		
		MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
Fixed	RGB	3,64998	0,99888	42,90676	4,33561	0,99877	42,08716	5,76854	0,99828	40,72572
Fixed	R	1,59278	0,99861	46,33022	1,74654	0,99863	46,06134	2,41306	0,99845	44,49186
Fixed	G	1,98915	0,99859	45,23842	2,64453	0,99839	44,01267	2,92667	0,99845	43,56076
Fixed	B	3,17231	0,99825	43,27773	3,08337	0,99838	43,31315	4,22454	0,99811	41,95068
Shifting	RGB	5,73314	0,99871	41,15536	5,17263	0,99876	41,66821	5,82260	0,99850	40,66728
Shifting	R	2,33284	0,99867	44,81792	2,04342	0,99859	45,45686	2,45993	0,99853	44,36571
Shifting	G	2,36345	0,99867	44,47459	2,48055	0,99863	44,51597	2,96048	0,99855	43,63509
Shifting	B	2,08455	0,99862	45,25081	2,37365	0,99857	44,73196	3,69669	0,99830	42,83457
Rotating	RGB	4,27077	0,99878	42,03912	3,97795	0,99882	42,46192	6,02282	0,99826	40,58824
Rotating	R	1,87526	0,99857	45,58983	1,75926	0,99860	46,00693	2,52313	0,99844	44,34743
Rotating	G	2,27272	0,99861	44,78375	2,68847	0,99848	43,86408	3,53121	0,99843	42,67002
Rotating	B	2,80917	0,99842	43,81271	2,81857	0,99848	43,75436	3,94079	0,99821	42,37481
Shifting-Rotating	RGB	4,48060	0,99884	41,89339	5,43900	0,99860	41,05086	6,12345	0,99836	40,54377
Shifting-Rotating	R	1,92428	0,99871	45,48971	2,29056	0,99861	44,76303	2,57628	0,99850	44,24669
Shifting-Rotating	G	2,37695	0,99858	44,43567	2,65926	0,99857	43,98701	3,12673	0,99850	43,25716
Shifting-Rotating	B	3,23867	0,99850	43,64457	2,91100	0,99855	43,92567	3,73666	0,99833	42,62338

Table 9. Quality Metrics Results of proposed data hiding algorithms and KBM for 2nd Secret Message 16x16

Block Structure	Data Hidden Colour Channel	Sequential Data Hiding			Sequential Data Hiding in All Frames			Non-Sequential Data Hiding in All Frames		
		MSE	SSIM	PSNR	MSE	SSIM	PSNR	MSE	SSIM	PSNR
Fixed	RGB	4,84558	0,99819	42,19664	4,61092	0,99843	41,94859	0,10958	0,99983	59,58094
Fixed	R	3,18423	0,99674	43,26554	3,27562	0,99657	43,07835	0,10466	0,99961	58,02032
Fixed	G	4,73801	0,99668	41,49560	5,38541	0,99649	41,10626	0,32292	0,99959	54,08266
Fixed	B	5,37134	0,99703	41,06739	6,25932	0,99664	40,33113	0,63023	0,99954	51,57044
Shifting	RGB	3,07594	0,99894	45,05378	3,82382	0,99872	42,94331	0,08092	0,99991	62,14244
Shifting	R	1,46701	0,99829	46,61326	1,31024	0,99835	47,20597	0,04970	0,99980	61,19991
Shifting	G	4,00405	0,99794	42,42346	4,14305	0,99799	42,10196	0,25976	0,99973	55,87906
Shifting	B	3,87630	0,99786	42,54503	4,86245	0,99780	41,42280	0,36580	0,99971	54,37389
Rotating	RGB	2,86662	0,99902	44,47779	2,44849	0,99906	44,98394	0,09181	0,99988	61,68752
Rotating	R	1,92822	0,99767	45,33455	1,76925	0,99760	45,79728	0,07534	0,99973	59,73034
Rotating	G	3,42685	0,99754	43,15970	3,08787	0,99751	43,53908	0,19929	0,99969	56,30115
Rotating	B	5,26540	0,99735	40,98734	5,01514	0,99748	41,24836	0,51274	0,99965	51,70028
Shifting-Rotating	RGB	2,51485	0,99903	45,11638	1,87244	0,99912	46,44285	0,04773	0,99992	64,50966
Shifting-Rotating	R	1,65209	0,99808	46,15068	1,30096	0,99815	47,13003	0,05956	0,99975	60,61912
Shifting-Rotating	G	2,50704	0,99827	44,60792	2,41035	0,99836	45,18766	0,22134	0,99976	56,53756
Shifting-Rotating	B	5,53189	0,99786	41,03881	5,42036	0,99784	41,14083	0,42605	0,99971	54,25784

The quality metrics obtained from the results of sequential data hiding operations performed without using KBM according to the 4x4, 8x8 and 16x16 DCT block structure of the 2nd secret message into the stego object are presented in Table 6 whereas the results of data hiding operations using proposed data hiding algorithms and KBM are shown in Table 7, Table 8, and Table 9.

5. Discussion

For the 1st secret message;

It's seen that the best quality metric values for each DCT block structure in data hiding operations without using KBM are obtained when the R colour channel is selected for data embedding (Table 3 to Table 5). The best quality metric values obtained in data hiding processes using KBM and proposed data hiding algorithms for different DCT blocks are as follows:

- For 4x4 DCT block structure; rotation & shifting key block structure and the non-sequential data-hiding in all frames algorithm,
- For 8x8 DCT block structure; rotation key block structure and the non-sequential data-hiding in all frames algorithm,

For 16x16 DCT block structure; rotation & shifting key block structure and the non-sequential data-hiding in all frames algorithm.

For the 2nd secret message;

It's again seen that the best quality metric values for each DCT block structure in data hiding operations without using KBM are obtained when the R colour channel is selected for data embedding (Table 7 to Table 9). The best quality metric values obtained in data hiding processes using KBM and proposed data hiding algorithms for different DCT blocks are as follows:

- For 4x4 DCT block structure; shifting key block structure and the non-sequential data-hiding in all frames algorithm,
- For 8x8 DCT block structure; fixed key block structure and the sequential data-hiding algorithm,
- For 16x16 DCT block structure; rotation & shifting key block structure and the non-sequential data-hiding in all frames algorithm.

When data hiding operations by the proposed methods are compared according to DCT block structures for both messages;

- For "Sequential data hiding" and "Sequential data hiding in all frames" algorithms, the best quality metric values are obtained from applications using 4x4 DCT block structure,
- For the "Non-sequential data hiding in all frames" algorithm, the best quality metric values are obtained from applications using a 16x16 DCT block structure.

When the results obtained from data hiding operations performed with and without the proposed method using KBM are compared it's seen that the quality metrics obtained from the proposed methods using KBM are higher.

When the results obtained from data hiding operations performed with the proposed method using KBM for the 1st and the 2nd secret messages are considered; it's seen that the performance metrics values of 4x4 and 16x16 DCT block structures don't have significant changes whereas 8x8 DCT block structure has worse values when the size of data increases.

When the DCT block structure is small, more data can be hidden in the video frame and vice versa. Changes in MSE&PSNR values occur due to the change in the amount of data hidden in the video frame. If the amount of hidden data increases, it is found that MSE decrease and PSNR increases in the sequential data hiding algorithm. When non-sequential data-hiding in all frame algorithm considered MSE increases and PSNR decreases if the amount of the hidden data increases.

When the SSIM quality metric is considered, it's seen that the rate of change in image quality is quite small whatever the proposed algorithms are used. It means that structural similarity between the frames is very high and there is no noticeable change in the perceived visual quality of the image.

Data security is one of the most important issues of steganography and handled under the robustness parameter. KBM structures used for data hiding in the proposed methods increase the security of the hidden data. The positioning of the key blocks, the variable pattern structures, the selective placement of the areas to be hidden in the pattern, direction and state controls, shifting and rotating the pattern structure for each block increase the security of the hidden data for the proposed method.

No visual problems were encountered on the video frames during data hiding with the 8x8 and 16x16 pixel DCT block structure. However, in sequential data hiding performed with a 4x4 DCT block structure, it's seen that there is a slight pixel distortion on some video frames. It's suggested to use wither 8x8 or 16x16 pixel DCT blocks for the proposed method. The "non-sequential data hiding in all frames" algorithm gives the most successful result in terms of data security in line with the quality metrics obtained from the methods proposed for each DCT block structure.

6. Conclusion

The data is hidden in 4x4, 8x8 and 16x16 DCT blocks directly by using KBM with the proposed methods. Video frame structures were analyzed before and after the data hiding process in order to observe the changes in video frames and the results obtained were presented. No visual problems were encountered on the video frames during data hiding with the 8x8 and 16x16 pixel DCT block structure. However, in sequential data hiding performed with a 4x4 DCT block structure, it's seen that there is a slight pixel distortion on some video frames. It's suggested to use wither 8x8 or 16x16 pixel DCT blocks for the proposed method. The "non-sequential data hiding in all frames" algorithm gives the most successful result in terms of data security in line with the quality metrics obtained from the methods proposed for each DCT block structure. Additionally, the structure of the KBM (fixed, rotating, shifting, shifting&rotating) isn't the primary determiner and the one that gives the best results for the secret message must be used. This study tried to fill the gap for the security of data in video steganography. Future works of this study will be adding artificial intelligence based KBM and encryption methods to the proposed method to increase the security more.

References

[1] Patel, H. and P. Dave, Steganography technique based on DCT coefficients. International Journal of Engineering Research and Applications, 2012. 2(1): p. 713-717.

[2] Walia, E., P. Jain, and N. Navdeep, An analysis of LSB & DCT based steganography. Global Journal of Computer Science and Technology, 2010.

[3] D.Nithya Kalyani, D.K.M., Safe Information Hiding Using Video Steganography. International Journal of Computer Science and Mobile Computing, 2015. 4(7): p. 502-512.

[4] Yeşilyurt, M., Et Al., Sikiştirilmiş İmgeler İçin Dct Tabanli Görünmez Damgalama Uygulamasi A Dct Based Invisible Watermarking Application For Compressed Images.

[5] .Liu, Y., et al., Video steganography: A review. Neurocomputing, 2019. 335: p. 238-250.

[6] Goyal, S. and M.S. Nehra. Texture based video steganography technique using block-wise encryption. in 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). 2017. IEEE.

[7] Zhang, H., et al. Video steganography with perturbed macroblock partition. in Proceedings of the 2nd ACM workshop on Information hiding and multimedia security. 2014.

[8] Wibisurya, A., Distributed steganography using five pixel pair differencing and modulus function. Procedia computer science, 2017. 116: p. 334-341.

[9] Sonawane, R.A., et al., Reversible Texture Synthesis using Steganography. International Journal of Engineering Science, 2019. 20250.

[10] Ramalingam, M. and N.A.M. Isa, A data-hiding technique using scene-change detection for video steganography. Computers & Electrical Engineering, 2016. 54: p. 423-434.

[11] Prasad, S. and A.K. Pal, An RGB colour image steganography scheme using overlapping block-based pixel-value differencing. Royal Society open science, 2017. 4(4): p. 161066.

[12] Idbeaa, T., S. Abdul Samad, and H. Husain, A secure and robust compressed domain video steganography for intra-and inter-frames using embedding-based byte differencing (EBBD) scheme. PLoS one, 2016. 11(3): p. e0150732.

[13] UZBAS, B., Digital Dental Models Automatic Gender Detection in Institute of Science, Computer Engineering USA.2017, Selçuk University. p. 113.

[14] Akramullah, S., Digital Video Concepts, Methods and Metrics. 2014: Spres Open.

[15] Panasiuk, J. and L. Kowalczuk, On bitstream features based perceptual quality estimation of HEVC coded videos. 2015.

[16] Haas, D.M., et al., Association of Adverse Pregnancy Outcomes With Hypertension 2 to 7 Years Postpartum. J Am Heart Assoc, 2019. 8(19): p. e013092.