

## Research Article

# Coverless Steganography Based on Motion Analysis of Video

Yun Tan , Jiaohua Qin , Xuyu Xiang , Chunhu Zhang , and Zhangdong Wang 

College of Computer Science and Information Technology, Central South University of Forestry & Technology, 410004 Changsha, China

Correspondence should be addressed to Jiaohua Qin; qinjiaohua@163.com

Received 2 March 2021; Revised 25 March 2021; Accepted 30 March 2021; Published 22 April 2021

Academic Editor: Beijing Chen

Copyright © 2021 Yun Tan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of interactive multimedia services and camera sensor networks, the number of network videos is exploding, which has formed a natural carrier library for steganography. In this study, a coverless steganography scheme based on motion analysis of video is proposed. For every video in the database, the robust histograms of oriented optical flow (RHOOOF) are obtained, and the index database is constructed. The hidden information bits are mapped to the hash sequences of RHOOOF, and the corresponding indexes are sent by the sender. At the receiver, through calculating hash sequences of RHOOOF from the cover video, the secret information can be extracted successfully. During the whole process, the cover video remains original without any modification and has a strong ability to resist steganalysis. The capacity is investigated and shows good improvement. The robustness performance is prominent against most attacks such as pepper and salt noise, speckle noise, MPEG-4 compression, and motion JPEG 2000 compression. Compared with the existing coverless information hiding schemes based on images, the proposed method not only obtains a good trade-off between hiding information capacity and robustness but also can achieve higher hiding success rate and lower transmission data load, which shows good practicability and feasibility.

## 1. Introduction

In recent years, the demand for information hiding continues to grow, especially for cloud computing environments. Traditional information hiding technologies usually embed secret information in the carrier [1–5] and lead to variable modification of carrier features. The steganography schemes that hide information by constructing the mapping relationship between cover features and secret information [6, 7] or using autogeneration technology [8–10] have aroused the interest of many researchers, which have a strong ability to resist steganalysis.

Most existing coverless steganography schemes are based on text and images. The text-based methods dug out the text features such as Chinese numeral expression [11], word rank map [12], or word frequency [13, 14] and quantified them. Then, the mapping relationship between text features and secret information was established, and the indexes were constructed. While in the image-based methods, the key problem is how to extract the main features of an image efficiently, which have been extensively studied

in previous research studies [15–19]. In the method proposed by Zhou et al. [6], the secret information was converted to bits and divided into several data segments. The image with the same hash sequence as the data segment was selected and transmitted to the receiver as the cover image, from which the receiver could extract the secret information. Zheng et al. [20] used the direction information of scale-invariant feature transform (SIFT) points to design image hash and used the inverted index of quad-tree structure to improve the capacity and retrieval efficiency. An algorithm based on histograms of oriented gradients (HOG) was proposed by Zhou et al. [21], which obtained the hash sequences from the nonoverlapping blocks of the image. After block discrete cosine transformation (DCT) [22] or discrete wavelet transformation (DWT) [23], the relationship between coefficients of adjacent blocks was used to generate robust feature sequences. It can improve the capacity of hiding information by partitioning the image, but the robustness will be reduced by a larger partition number. Zou et al. [24] and Cao et al. [25] used average pixel values of subimages, which achieve a high hiding success rate and

capacity. In [26], LBP feature of the medical image was extracted and mapped to privacy information. Recently, Luo et al. [27] used recognized objects to hide secret information.

At the same time, live network platforms and video social applications are becoming more and more popular [28]. A large number of short videos have been generated and spread on the Internet, which provides sufficient carrier for information hiding. Compared with image, video not only has texture, shape, and color features but also has rich spatial and temporal features, from which some motion characteristics can be mined. Theoretically, motion characteristic is robust and cannot easily be tampered with, which is suitable for steganography. This motivates us to design a novel coverless steganography scheme based on short videos by constructing a mapping function between the motion characteristics and information bits.

Existing research results of coverless steganography based on videos are still rare. Some researchers have proposed some zero-watermarking technologies for copyright protection of video, which constructed watermark information by extracting video features. Li et al. [29] proposed a zero-watermarking algorithm based on logarithmic polar coordinate transformation. After 2D-DWT and 3D-DCT transformation of the original image, the zero-watermarking was realized by transforming the logarithmic polar coordinates. Liu et al. [30] proposed a zero-watermarking scheme for three-dimensional video, which extracted the features of two-dimensional video frames and depth maps to generate copyright information and used secret sharing schemes to achieve copyright protection. Compared with the zero-watermarking algorithms, coverless steganography based on videos has a higher requirement for capacity.

As shown in Figure 1, there is a baby walking in the video. The simplest mapping function is to connect the stepping of different feet with different information bits as the following equation:

$$f_{map} = \begin{cases} 0, & \text{action is "step left,"} \\ 1, & \text{action is "step right."} \end{cases} \quad (1)$$

Then, the continuous stepping of the baby can represent a sequence of information bits. However, this kind of mapping has some shortcomings: first, the stepping characteristic is semantic and can easily be understood and cracked. Second, the calculation complexity of stepping recognition is still relatively high, although the motion analysis and tracking of a video have achieved significant progress recently [31, 32]. Third, the capacity of information hiding is low since only one bit is hidden in every frame. Therefore, how to mine the nonsemantic motion characteristic and construct a correlated mapping function for information bits is the key issue of coverless steganography based on videos.

Video recognition has been studied in depth by a lot of researchers, and many algorithms have been proposed. Optical flow is the most classical method for video analysis [33–37]. In this work, we mainly study the optical flow characteristics of video and map them with hidden information to realize coverless steganography. The main

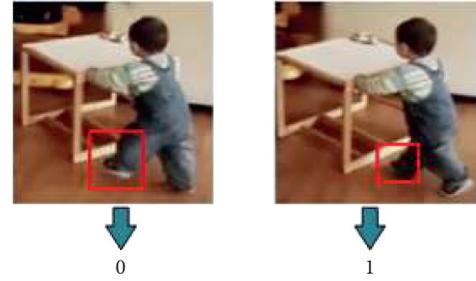


FIGURE 1: Mapping between motion characteristic and information bit.

contributions of this work are as follows: first, we construct a novel coverless steganography scheme based on motion analysis of video. Second, the mapping algorithm between the robust directional characteristics of video optical flow and secret information is proposed and optimized. Finally, information hiding capacity, robustness performance, efficiency, hiding success rate, and transmission data load of the proposed scheme are analysed and compared with the existing coverless steganography schemes based on images.

The study is organized as follows: preliminaries are introduced in the second section, and the proposed method is described in the third section. Experimental results and comparisons are shown in the fourth section. Finally, we conclude this study in the last section.

## 2. Preliminaries

**2.1. Optical Flow.** Optical flow is the instantaneous velocity distribution of the brightness pattern, which is caused by the movements of objects [33, 34] and has been applied widely for motion analysis. In recent studies, the optical flow was used to estimate the traffic flow parameters of moving vehicles in different scenarios and shew effectiveness [35, 36]. Lv et al. realized subpixel image registration based on the optical flow model and feature-point matching [37].

The basic idea of the optical flow method is to find the corresponding relationship between adjacent frames in the image sequence using the change of pixels in time domain. Then, the motion information of objects can be calculated.

Assuming there is a pixel  $(x, y, t)$  in one frame, its light intensity is expressed as  $I(x, y, t)$ . It moves to  $(x', y', t')$  in the next frame. According to the assumption of constant brightness, we can get the following equation:

$$I(x, y, t) = I(x', y', t') = I(x + dx, y + dy, t + dt). \quad (2)$$

The Taylor series approximation is applied to equation (2); then, we can get

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + \varepsilon, \quad (3)$$

where  $\varepsilon$  is the second-order infinitesimal term and can be neglected. Therefore, equation (3) can be transformed to

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0. \quad (4)$$

Assuming  $u$  and  $v$  are the velocity vectors of optical flow along the  $X$  axis and  $Y$  axis, respectively, we have

$$\begin{aligned} u &= \frac{dx}{dt}, \\ v &= \frac{dy}{dt}. \end{aligned} \quad (5)$$

Equation (4) can be transformed as

$$I_x u + I_y v + I_t = 0, \quad (6)$$

where  $I_x = \partial I / \partial x$ ,  $I_y = \partial I / \partial y$ , and  $I_t = \partial I / \partial t$ . This is the basic constraint equation of optical flow.

In order to solve the above equations and achieve the value of the unknown  $u$  and  $v$ , there are two classical methods. One is a global differential method, which assumes that the optical flow changes smoothly over the entire image. The other is a local differential method, which assumes that the motion vector remains constant over a small spatial domain. Therefore, it is suitable for small motion detection, but fails for large motion detection. In order to improve this defect, the pyramidal implementation was proposed by Bouguet [38].

The pyramid layering method was used to reduce the size of the image layer by layer, thereby reducing the motion displacement of the object between two frames. The process is shown in Figure 2, and the specific steps are as follows:

Step1: A pyramid is created for every frame, and the resolution is sequentially lowered from the bottom to the top.

Step2: Starting at the top level, the optical flow at every point in the top-level image is obtained by minimizing the minimum matching error sum within the neighbourhood of each point. Assuming  $d$  is the optical flow, the residual function is defined as [38]

$$\begin{aligned} \varepsilon(d_x, d_y) &= \varepsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \\ &\cdot \left( I(x, y) - J(x + d_x, y + d_y) \right)^2, \end{aligned} \quad (7)$$

where  $(u_x, u_y)$  is the point of the original image  $I$  and  $(u_x + d_x, u_y + d_y)$  is the point of the target image  $J$ .

Supposing there are  $L$  layers of the pyramid, the first layer is the original image. If the total displacement is  $d$ , then the displacement for each layer is

$$d^L = \frac{d}{2^L}. \quad (8)$$

Step 3: The optical flow of the layer  $L$  is propagated to the layer  $L - 1$  as follows:

$$g^{L-1} = 2(g^L + d^L). \quad (9)$$

For the layer  $l$ , the calculation of optical flow is based on the minimization of the sum of matching error for all points in the neighbourhood area, as the following equation:

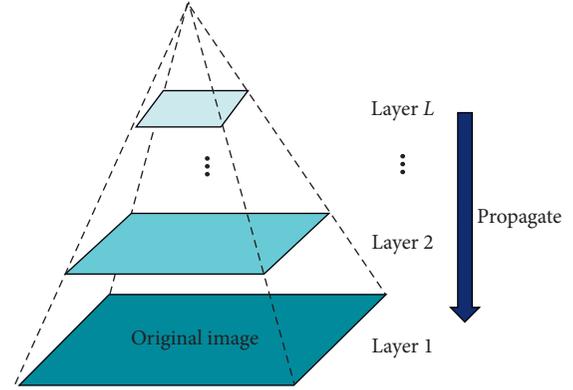


FIGURE 2: Pyramid layering process.

$$\begin{aligned} \varepsilon^l(d^l) &= \varepsilon^l(d_x^l, d_y^l) = \sum_{x=u_x^l-w_x}^{u_x^l+w_x} \sum_{y=u_y^l-w_y}^{u_y^l+w_y} \\ &\cdot \left( I^l(x, y) - J^l(x + g_x^l + d_x^l, y + g_y^l + d_y^l) \right)^2. \end{aligned} \quad (10)$$

It is propagated down the pyramid until it reaches the bottom layer. Then, the optical flow is calculated by

$$d = g^1 + d^1. \quad (11)$$

**2.2. Robust Histogram of Oriented Optical Flow.** Since the size of the moving target usually changes with time in a video, the dimension of the corresponding optical flow descriptor will also change. At the same time, the original optical flow is also sensitive to the background noise, scale change, and the direction of motion. For information hiding, the extracted features are expected to be more stable, which can gain better robustness. Therefore, it is necessary to find a method based on the optical flow that can not only characterize the temporal motion information but also be insensitive to scale. Histogram of oriented optical flow (HOOF) was proposed by Chaudhry et al. [39]. The scale invariance of HOOF feature was achieved by the normalized histogram. In order to further enhance the robustness, the robust histogram of oriented optical flow (RHOOF) is achieved by only counting the number of optical flows located in the directional bins, while the amplitude information is ignored. This means that RHOOF will not be affected by the amplitude variation of optical flow, which is different from the original HOOF.

For every two frames, the optical flow is calculated. And then, the directional angle of the optical flow vector can be achieved by

$$\theta = \text{atan2}\left(\frac{y}{x}\right), \quad (12)$$

where  $\text{atan2}(\cdot)$  is a four-quadrant inverse tangent function,  $x$  is the horizontal component, and  $y$  is the vertical component of optical flow vector. The range of  $\theta$  is  $[-\pi, \pi]$ . If the angle range is divided into several groups, then the histogram distribution is statistically obtained. As shown in Figure 3,

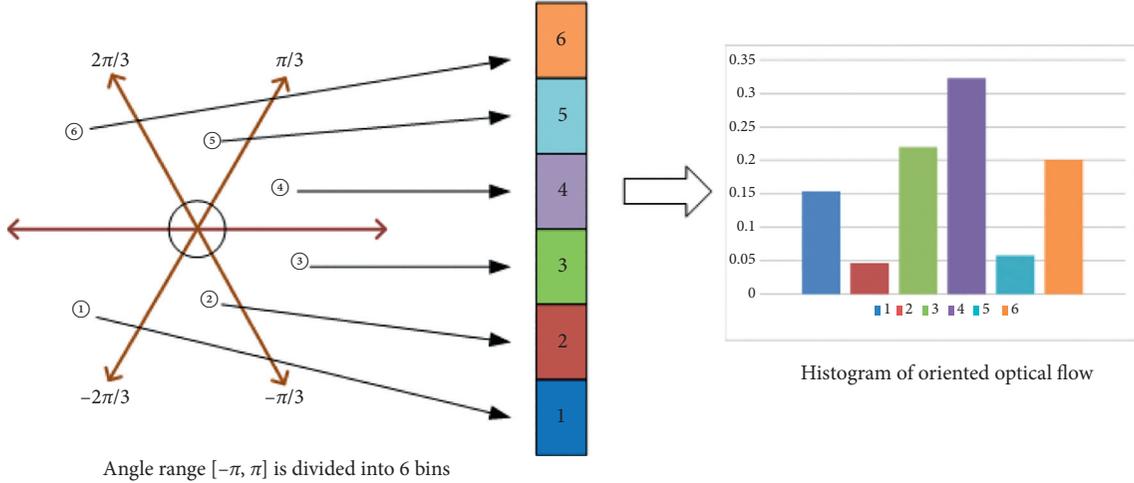


FIGURE 3: The principle of the histogram of oriented optical flow.

the bin number of the histogram is 6 and the bin size is  $\pi/3$ , which means that the angle range of optical flow is divided to 6 groups. The distribution of the angles is shown in the histogram on the right. The sum of the possibilities of all groups is 1.

### 3. Proposed Coverless Steganography Scheme

Our proposed coverless steganography scheme based on videos is shown in Figure 4. The framework mainly includes three parts: index construction, secret information hiding, and secret information extraction. First, the video database is composed of multiple videos with different topics. The video database is shared by both secret information sender and receiver, which can be stored on cloud platform to save the storage space of the end user. Second, calculate RHOOF for every video in the video database. After the hash sequences of RHOOF are calculated, the video index database is constructed. The construction of video database and index database is the basis of coverless information hiding.

During the information hiding process, the secret information needs to be preprocessed and divided to binary bit groups. Every bit group can be mapped to a hash sequence of RHOOF. After searching in the video index database, the appropriate one or several videos are selected as cover, and the corresponding mapping indexes are returned to the sender. The mapping indexes will be sent to the receiver. At the receiver, cover video can be found accurately and efficiently according to the received mapping indexes. Through calculating the hash sequences of RHOOF from the cover video, the secret information can be recovered successfully. During the whole process, the cover video remains original without any modification. Therefore, it can resist the detection of steganalysis.

**3.1. Generation of Hash Sequence.** As described previously, RHOOF can reflect the main movement characteristic of the video. We propose a hash sequence generation method based on RHOOF as shown in Figure 5. For the two adjacent

frames (assuming as frame  $i$  and frame  $i + 1$ ) of a video, they are transformed to gray scale first. Second, the two frames are median filtered in order to suppress the possible noise and protect the edge information. Third, the pixel changes of these two frames are calculated, and the oriented optical flows are achieved as described in the previous section. We can analyse the orientation values of optical flow. The histogram is calculated in several bins for every subblock. In Figure 5, we set the number of subblocks to 4 and the number of bins to 8 as an example.

Assuming the histogram is denoted as

$$H = \{h_1, h_2, \dots, h_N\}, \quad (13)$$

where  $N$  is the number of the bins. We set the threshold as

$$\text{thres} = \sum_{i=1}^N \frac{h_i}{N} + \text{th}_0, \quad (14)$$

where  $\text{th}_0$  is a correction factor. Then, the hash sequence  $b_1 b_2 \dots b_N$  is achieved by comparing the histogram value with the threshold as the following equation:

$$b_i = \begin{cases} 1, & \text{if } h_i \geq \text{thres}, \\ 0, & \text{if } h_i < \text{thres}, \\ 1 \leq i \leq N. \end{cases} \quad (15)$$

**3.2. Construction of Video Index Database.** In order to find the cover video efficiently and accurately, the construction of video index database is necessary and important. Therefore, we construct an efficient index database with two levels as shown in Figure 6. The first level index is the hash sequence and the second level index contains the information items of cover video and cover frame.

The index items are sorted by the hash sequence. Here, the bin number is also set to 8 as an example. Therefore, the value of hash sequence varies from "00000000" to "11111111" in the index table. For every index item corresponding to the hash sequence, index ID, video ID, frame

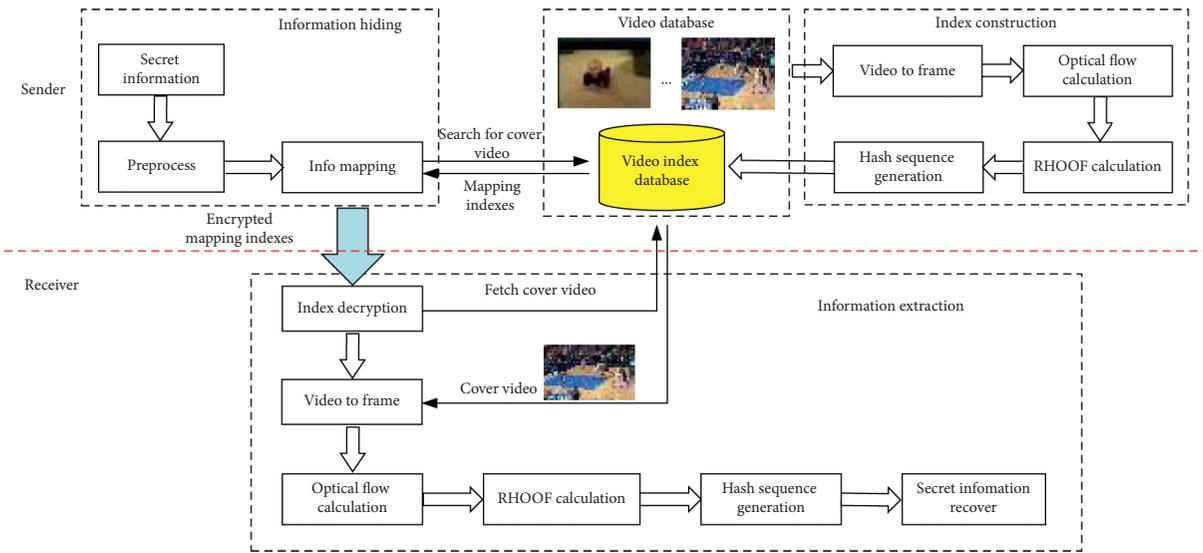


FIGURE 4: The framework of the proposed coverless steganography scheme.

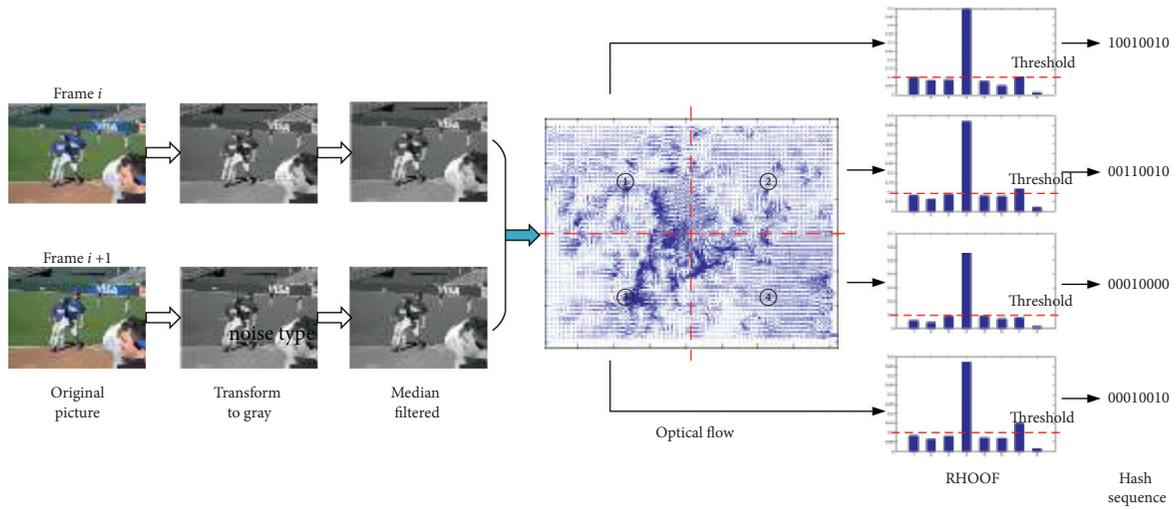


FIGURE 5: The process of hash sequence generation.

Hash sequence	Index ID	Video ID	Frame ID	Subblock ID
00000000	1	\Video\walking.avi	120	2
00000001	2	\Video\running.avi	10	3
...	...	...	...	...
00000010	8	\Video\basketball.avi	145	1
...	...	...	...	...
...	9	\Video\walking.avi	23	4
...	10	\Video\walking.avi	167	2
...	...	...	...	...
...	13	\Video\crwaling.avi	40	3
...	...	...	...	...
11111111	902	\Video\walking.avi	220	2
...	903	\Video\boxing.avi	30	1
...	...	...	...	...
...	1000	\Video\crwaling.avi	78	4

FIGURE 6: Structure of video index database.

ID, and subblock ID are contained in the index database. Index ID is the serial number of an index, which is incremental. Video ID means the storage path and the name of the cover video. Frame ID means the corresponding frame, and subblock ID means the corresponding subblock. With such information, the cover video can be found accurately, and the hash sequence can be calculated efficiently and conveniently. For one hash sequence, it is possible to contain multiple index items, which means there are multiple cover subblocks with the same hash sequence in the video database. In this situation, any index item can be chosen in principle for subsequent secret information mapping.

**3.3. Secret Information Hiding.** During secret information hiding, how to map the secret information to the cover video efficiently is the most critical part. The whole process is summarized as follows:

Step 1: construct a video database, which is shared by both the sender and the receiver

Step 2: for each video in the library, the frame optical flows are obtained as described in the previous section

Step 3: the directional angle of the optical flow vector is calculated by equation (12)

Step 4: for every frame, the robust histogram distribution is statistically counted in subblocks based on the oriental information of optical flow. The hash sequences are obtained as described previously.

Step 5: construct the video index database as described previously

Step 6: the secret information needs to be preprocessed before sending. Assuming the length of the secret information is  $k$  bits, it will be divided into  $m$  segments as

$$m = \lceil \frac{k}{N} \rceil, \quad (16)$$

where  $N$  is the bin number of RHOOF statistics. For the last segment, "0" bits are padded to the tail, and the number of the padding bits is

$$p = \lceil \frac{k}{N} \rceil \cdot N - k. \quad (17)$$

Step 7: for every segment, we search the corresponding index item in the index database, which has the hash sequence equal to the information bits. It is possible that there are multiple index items mapping to the same hash sequence. In order to increase the efficiency of information extraction, we should choose the index items with the same video file as much as possible. For the same video file, the mapping index item with smaller index ID will be chosen.

Step 8: the information of mapping indexes corresponding to the secret information segments will be sent to the receiver. In order to enhance security, the index information can be encrypted before transmission.

The detailed algorithm of index database construction is described in Algorithm 1.

The information hiding algorithm at the sender is described in Algorithm 2.

**3.4. Extraction of Secret Information.** At the receiver, by calculating the hash sequence of RHOOF based on the cover video, the secret information can be extracted successfully. The process of secret information extraction is as follows:

Step 1: after receiving the index information sent by the transmitter, the receiver will decrypt it first if necessary. Then, the index items will be analysed, and video ID, frame ID, and subblock ID can be obtained.

Step 2: the corresponding frame can be found according to video ID and frame ID. The optical flow of the corresponding frame is calculated as described previously.

Step 3: the directional angle of the optical flow vector is calculated by equation (12)

Step 4: the robust histogram distribution is statistically counted in subblocks according to the oriented information of the optical flow. The hash sequence is obtained.

Step 5: repeat steps 1–4 until all the hash sequences corresponding to the mapping index items have been extracted

Step 6: after connecting the hash sequences and removing the padding bits from the tail, the bitstream of secret information is recovered successfully

The detailed algorithm of secret information extraction is described in Algorithm 3.

As shown in Figure 7, there is an example of transmission and extraction of the secret bitstream as "1111111000000." The bin number  $N$  is 8. As mentioned before, the length of padded bitstream should be an integral multiple of  $N$ . Therefore, the bitstream is padded with one "0" bit at the tail first, which makes the length of bitstream as 16. Then, the padded bitstream is segmented to 2 groups of 8 bits. Next, the hash sequences equal to the segmented bitstream "11111111" and "00000010" are searched in the video index database. From Figure 6, it can be seen that there are multiple index items corresponding to the hash sequences "11111111" and "00000010." The index items with index ID as 902 and 9 are selected since they have the same video ID, which are marked with green as shown in Figure 6. Therefore, the video "walking.avi" is our cover video. At the receiver, RHOOF are calculated based on frame 220 and 221 and frame 23 and 24 of "walking.avi." Then, the hash sequences "11111111" and "00000010" are achieved from the subblock 2 and the subblock 4 separately. After removing the padding bit "0" from the tail, the secret bitstream "11111110000001" can be recovered successfully.

**3.5. Algorithm Improvement.** One prerequisite of the optical flow method is that the brightness should remain constant.

Input: video database  $V = \{V_1, V_2, \dots, V_d\}$ , number of videos  $d$ , smoothing window size for optical flow calculation  $W$ , number of LK pyramid level  $L$ , and bin number  $N$ .

Output: video index database  $I = \{\text{Ind}_1, \dots, \text{Ind}_{2^N}\}$

- (1) For  $i = 1 : d$
- (2) Decompose video to pictures:  $P = \text{VideoToFrame}(V_i)$
- (3) For  $j = 1 : \text{FrameNum} - 1$
- (4) Convert RGB to gray:  $R_j = \text{Rgb2gray}(P_j)$ ,  $R_{j+1} = \text{Rgb2gray}(P_{j+1})$
- (5) Median filtering:  $M_j = \text{medfilt}(R_j)$ ,  $M_{j+1} = \text{medfilt}(R_{j+1})$ ,
- (6) Calculate the hierarchical optical flow matrixes between frame  $j$  and  $j + 1$ :  $(u, v) = \text{HierarchicalLK}(M_j, M_{j+1}, W, L)$
- (7) Calculate RHOOF and hash sequences in subblocks  
For  $s = 1 : \text{subblock\_num}$
- (8)  $H_s = \text{RHOOF}(u, v, N)$
- (9)  $\text{Hash}_s = \text{HashCalc}(H_s)$
- (10) Link MySQL and update index database: index item  $\rightarrow$  {index ID, video ID, frame ID, subblock ID}
- (11) End for
- (12) End for
- (13) End for

ALGORITHM 1: Index database construction.

Input: video index database  $I = \{\text{Ind}_1, \dots, \text{Ind}_{2^N}\}$ , secret information bitstream  $B = \{b_1, b_2, \dots, b_k\}$

Output: mapping index set  $I_m = \{\text{Ind}_1, \dots, \text{Ind}_m\}$ , number of index items  $m$

- (1) Padding the secret information bitstream:  $B' = \text{pad}(B)$
- (2) Divide  $B'$  to  $m$  segments
- (3) For  $i = 1 : m$
- (4) Search the corresponding index item and update the index set as  $I_m = \{I_m, \text{Ind}_i\}$
- (5) End for
- (6) Send the mapping index set  $I_m$  to the receiver

ALGORITHM 2: Information hiding.

Input: video database  $V = \{V_1, V_2, \dots, V_d\}$ , smoothing window size for optical flow calculation  $W$ , number of LK pyramid level  $L$ , bin number  $N$ , mapped index set  $I_m = \{\text{Ind}_1, \dots, \text{Ind}_m\}$ , and number of index items  $m$

Output: secret information bitstream  $B = \{b_1, b_2, \dots, b_k\}$

- (1) For  $i = 1 : m$
- (2) Get video ID, frame ID, and subblock ID from the index item  $i$
- (3)  $P = \text{VideoToFrame}(V_{\text{video\_ID}})$
- (4) Convert RGB to gray:  $R_1 = \text{Rgb2gray}(P_{\text{Frame\_ID}})$ ,  $R_2 = \text{Rgb2gray}(P_{\text{Frame\_ID} + 1})$ ,
- (5) Median filtering:  $M_1 = \text{medfilt}(R_1)$ ,  $M_2 = \text{medfilt}(R_2)$
- (6) Calculate the hierarchical optical flow matrixes between corresponding frames:  $(u, v) = \text{HierarchicalLK}(M_1, M_2, W, L)$
- (7) Calculate the gradient histogram of optical flow of the corresponding subblock:  $H_{\text{Subblock\_ID}} = \text{gradientHist}(u, v, N)$
- (8)  $H_s = \text{HashCalc}(H_{\text{Subblock\_ID}})$
- (10) End for
- (11) Connect all the segments as  $\{H_{s_1}, H_{s_2}, \dots, H_{s_m}\}$
- (12) Remove padding bits, and the secret information bitstream is recovered:  $B = \{b_1, b_2, \dots, b_k\}$

ALGORITHM 3: Information extraction.

In the case of noise or random interference, the optical flow value will be greatly affected, which will lead to wrong extraction of the hidden information. Therefore, the algorithm is further optimized with an averaging window applied. Before the optical flow is calculated, we update the data of every frame by averaging the pixel values of adjacent frames.

Through this smoothing operation, the influence of noise and random interference can be reduced. The improved algorithm of index database construction is described in Algorithm 4.

The improved information extraction algorithm is described in Algorithm 5.

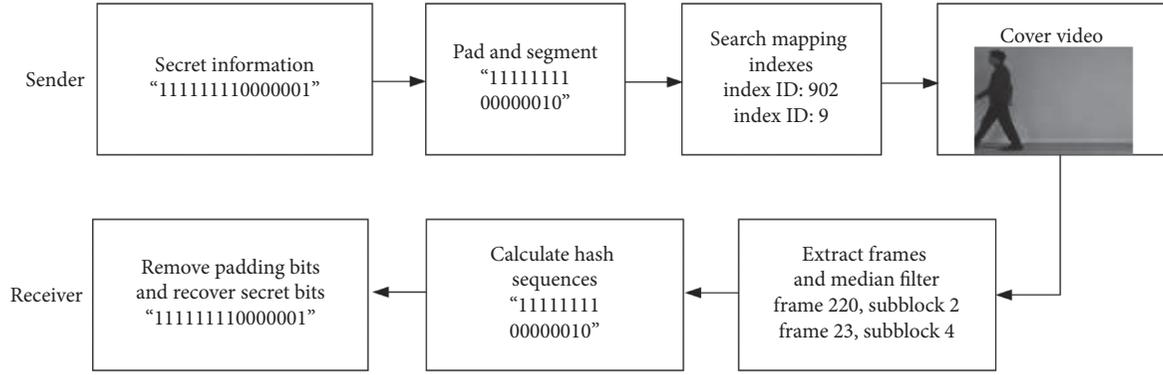


FIGURE 7: Example of the secret information transmission and extraction.

Input: video database  $V = \{V_1, V_2, \dots, V_d\}$ , number of the videos  $d$ , frame averaging window length  $avg\_L$ , smoothing window size for the optical flow calculation  $W$ , number of the LK pyramid level  $L$ , and bin number  $N$   
 Output: video index database  $I = \{Ind_1, \dots, Ind_{2^N}\}$

- (1) For  $i = 1 : d$
- (2) Decompose video to pictures:  $P = \text{VideoToFrame}(V_i)$
- (3) For  $j = 1 : \text{FrameNum} - avg\_L + 1$
- (4) Initialize  $R\_sum$  as all zero
- (5) For  $k = 0 : avg\_L - 1$
- (6) Convert RGB to gray:  $orig\_R_{j+k} = \text{Rgb2gray}(P_{j+k})$
- (7)  $R\_sum = R\_sum + orig\_R_{j+k}$
- (8) End for
- (9)  $R_j = R\_sum / avg\_L$
- (10) End for
- (11) For  $j = 1 : \text{FrameNum} - avg\_L$
- (12) Repeat step 1–10 of Algorithm 1.
- (13) End for
- (14) End for

ALGORITHM 4: Improved index database construction.

Input: video database  $V = \{V_1, V_2, \dots, V_d\}$ , frame averaging window length  $avg\_L$ , smoothing window size for optical flow calculation  $W$ , number of LK pyramid level  $L$ , bin number  $N$ , mapped index set  $I_m = \{Ind_1, \dots, Ind_m\}$ , and number of index items  $m$   
 Output: secret information bitstream  $B = \{b_1, b_2, \dots, b_k\}$

- (1) For  $i = 1 : m$
- (2) Get video ID, frame ID, and subblock ID from the index item  $i$
- (3)  $P = \text{VideoToFrame}(V_{\text{video ID}})$
- (4) For  $j = 1 : \text{FrameNum} - avg\_L + 1$
- (5) Initialize  $R\_sum$  as all zero
- (6) For  $k = 0 : avg\_L - 1$
- (7) Convert RGB to gray:  $orig\_R_{j+k} = \text{Rgb2gray}(P_{j+k})$
- (8)  $R\_sum = R\_sum + orig\_R_{j+k}$
- (9) End for
- (10)  $R_j = R\_sum / avg\_L$
- (11) End for
- (12) Median filtering:  $M_1 = \text{medfilt}(R_{\text{Frame ID}})$ ,  $M_2 = \text{medfilt}(R_{\text{Frame ID}+1})$
- (13) Repeat step 6–8 of Algorithm 3.
- (14) End for
- (15) Connect all the segments as:  $\{Hs_1, Hs_2, \dots, Hs_m\}$
- (16) Remove padding bits and the secret information bitstream is recovered:  $B = \{b_1, b_2, \dots, b_k\}$

ALGORITHM 5: Improved information extraction.

## 4. Experimental Results and Analysis

The experiments are conducted with the Intel(R) Core (TM) i7-6500X CPU @ 2.50 GHz and 16.00 GB RAM. Matlab 2018b is used for algorithm simulation, and MySQL workbench 6.3 is used for the index database construction.

A video database is used for the test, which is composed by videos with different movements and scenarios that are randomly chosen from UCF101 and HMDB51 datasets as shown in Figure 8. The file size of the videos is about 200~800 KB, and the duration time is about 2~10 seconds.

**4.1. Capacity Analysis.** The bit number of the generated hash sequence based on the cover video determines the capacity of information hiding. Assuming the frame number of the video is  $F$ , the number of optical flow images should be  $F - 1$ . Every optical flow image will be divided into  $S$  subblock, and the oriented histogram are statistically counted in  $N$  bins. Therefore, for every optical flow image, the number of mapped bits is:

$$C = S \cdot N. \quad (18)$$

Then for every video, the number of mapped bits is:

$$C' = S \cdot N \cdot (F - 1). \quad (19)$$

It can be seen that the capacity of information hiding in our scheme is related to the bin number  $N$ , subblock number  $S$ , and frame number  $F$ . For a specific video, the frame number is fixed, and then, the capacity is determined by the number of bins and subblocks. The larger the  $N$  and  $S$  are, the larger the capacity is. We compare the capacity of single optical flow image in our scheme with existing coverless information hiding schemes based on single image in Table 1. Here, in the proposed method, the subblock number is set as 4 and the bin number is set as 8.

For a secret message to be hidden, if the capacity of single image is larger, a smaller number of cover images will be needed. Assuming the length of the secret information is  $K$  bits, the capacity of single image is  $C$  bits; then, the number of required cover images is

$$M_{\text{image}} = \lceil \frac{K}{C} \rceil. \quad (20)$$

With the same hidden information, the number of images needed for different methods is compared in Table 2. It can be seen that our proposed method (set  $S = 4$ ,  $N = 8$ ) has a larger capacity than other methods. With the increment of  $S$  and  $N$ , the capacity will be even enlarged more. However, if  $N$  is increased, the size of video database needs also to be enlarged in order to ensure the success rate of information mapping. And the robustness will also be affected by the variation of  $S$  and  $N$ , which will be further investigated in the next tests.

**4.2. Robustness Analysis.** We investigate the robustness against pepper and salt noise, Gauss noise, and speckle noise with different parameters for performance evaluation. For a

video, the compression transformation is used commonly. Therefore, the effect of compressed MPEG-4 transformation (.mp4 file) and compressed motion JPEG 2000 file transformation (.mj2 file) are also investigated. Assuming the original bitstream is  $b_1 b_2 \dots b_m$  and the extracted bit sequence is  $b'_1 b'_2 \dots b'_m$ , the accuracy rate is calculated by

$$ACC = \frac{\sum_{i=1}^m f(i)}{m}, \quad (21)$$

where

$$f(i) = \begin{cases} 1, & \text{if } b_i = b'_i, \\ 0, & \text{if } b_i \neq b'_i. \end{cases} \quad (22)$$

The results of accuracy rate against different types of attacks with different bin number  $N$  are shown in Table 3, in which the subblock number  $S$  is fixed as 4. It can be seen that the robustness of the proposed scheme is good, especially against salt and pepper noise and MPEG-4 compression. At the same time, the increment of bin number will lead to the decrease of accuracy. It is because the smaller bin number will cause bigger bin size, which is less sensitive to the variation of angle distribution. But according to equation (19), the capacity will be decreased with the lower bin number.

The accuracy results with different types of attacks and different subblock number  $S$  are also investigated as given in Table 4. Here, the bin number  $N$  is fixed as 8. It can be seen that the effect of subblock number  $S$  is relatively small and the variation trend is irregular. The reason is that the spatial distribution of the optical flow is different for various types of videos. Therefore, according to equation (19), we can increase the capacity of information hiding by increasing the subblock number if necessary.

**4.3. Improvement Analysis of Frame Averaging.** We investigate the performance improvement of introducing the frame averaging window before calculating optical flow. Here, the length of frame averaging window is set as 10, which means that the data of every frame are updated by averaging the pixel values of adjacent 10 frames. The comparison of accuracy rate with different types of noises and compression is shown in Figures 9–12, where the subblock number is set as 4 and the bin number is set as 4, 8, 12, and 16, respectively.

It can be seen that frame averaging operation can improve the robustness significantly. Figure 9 is the accuracy comparison with compression attacks, which shows that the improvement for mj2 compression is much larger than for mp4 compression. For mj2 compression, the accuracy rate is increased by more than 7% with any bin number, while there is only weak improvement for mp4 compression. Figures 10 and 11 show the accuracy comparison with Gaussian noise and speckle noise, respectively, in which both have significant improvement. The increment of accuracy rate is even bigger when the bin number is increased. However, for salt and pepper noise, the accuracy rate will be decreased with frame averaging operation, as shown in Figure 12. And the

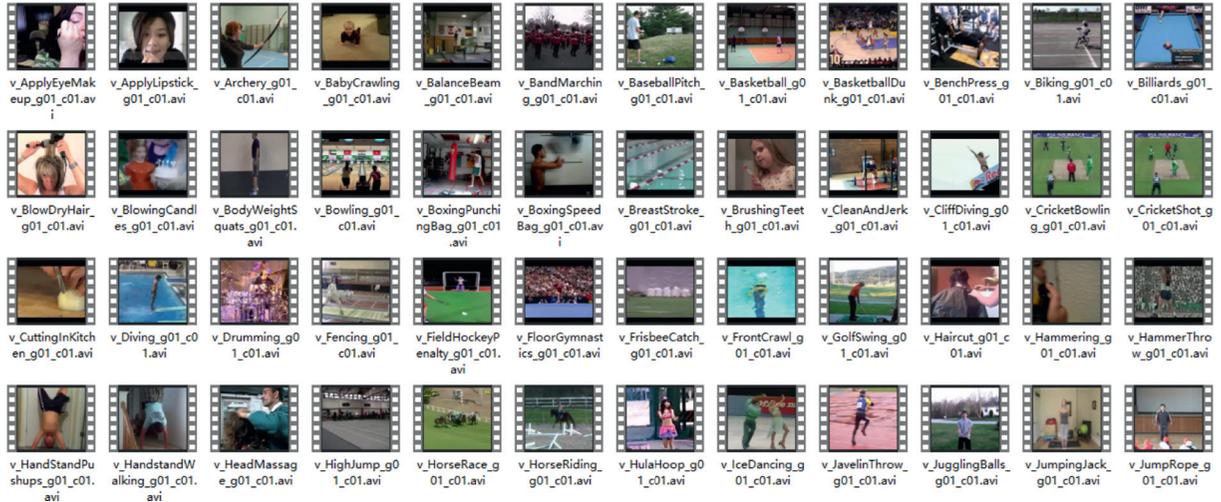


FIGURE 8: Video database.

TABLE 1: Capacity comparison.

Method	Pixel method [6]	Hash method [20]	DCT method [22]	DWT method [23]	Proposed method
Capacity	8 bit	18 bit	1~15 bit	1~15 bit	32 bit

TABLE 2: Number of needed images with the same hidden information.

Size of hidden information	1 B	10 B	100 B	1 kB
Pixel method [6]	1	10	100	1024
Hash method [20]	2	6	46	457
DCT method [22]	2~9	7~81	55~801	548~8193
DWT method [23]	2~9	7~81	55~801	548~8193
Proposed method	1	3	25	256

TABLE 3: Accuracy with different attacks and bin number  $N$ .

Attack	Accuracy			
	$N=4$	$N=8$	$N=12$	$N=16$
Salt and pepper ( $\sigma = 0.001$ )	0.9991	0.9986	0.9972	0.9950
Salt and pepper ( $\sigma = 0.005$ )	0.9964	0.9923	0.9890	0.9834
Salt and pepper ( $\sigma = 0.01$ )	0.9937	0.9877	0.9785	0.9686
Gauss ( $\sigma = 0.001$ )	0.8337	0.7005	0.6141	0.5133
Gauss ( $\sigma = 0.005$ )	0.8218	0.6485	0.5521	0.4438
Gauss ( $\sigma = 0.01$ )	0.8156	0.6198	0.5211	0.4104
Speckle ( $\sigma = 0.001$ )	0.8835	0.8235	0.7610	0.7037
Speckle ( $\sigma = 0.005$ )	0.8738	0.8098	0.7377	0.6743
Speckle ( $\sigma = 0.01$ )	0.8677	0.8000	0.7278	0.6596
Compressed MPEG-4 file with H.264 (.mp4 file)	0.9806	0.9589	0.9353	0.9126
Compressed motion JPEG 2000 file (.mj2 file)	0.8962	0.8476	0.8028	0.7599

larger the bin number is, the more obvious the impact is. This is because that salt and pepper noise is approximately equal in the amplitude but randomly distributed in different locations. Therefore, frame averaging calculation may possibly cause some clean pixels to be contaminated conversely.

**4.4. Robustness Comparison with Different Methods.** We use the methods based on images for performance comparison

after transferring the videos to frame images. The latest DWT method [22], DCT method [23], and Hash method [20] are considered and tested based on our video database. During tests, the subblock number of the DWT method and the DCT method is set as 8. The subblock number of our method is 4, and the bin number is also set as 4. The accuracy with Gaussian noise, salt and pepper noise, and video compression transformations in the different methods are shown in Figures 13–15. It can be seen that the proposed

TABLE 4: Accuracy with different attacks and subblock number.

Attack	Accuracy			
	S=1	S=2	S=4	S=8
Salt and pepper ( $\sigma = 0.001$ )	0.9980	0.9982	0.9986	0.9975
Salt and pepper ( $\sigma = 0.005$ )	0.9952	0.9952	0.9923	0.9930
Salt and pepper ( $\sigma = 0.01$ )	0.9896	0.9884	0.9877	0.9867
Gauss ( $\sigma = 0.001$ )	0.6583	0.6821	0.7005	0.7132
Gauss ( $\sigma = 0.005$ )	0.5994	0.6310	0.6485	0.6688
Gauss ( $\sigma = 0.01$ )	0.5695	0.6015	0.6198	0.6425
Speckle ( $\sigma = 0.001$ )	0.8176	0.8059	0.8235	0.8268
Speckle ( $\sigma = 0.005$ )	0.8074	0.7899	0.8098	0.8149
Speckle ( $\sigma = 0.01$ )	0.7934	0.7825	0.8000	0.8059
Compressed MPEG-4 file with H.264 (.mp4 file)	0.9665	0.9635	0.9589	0.9526
Compressed motion JPEG 2000 file (.mj2 file)	0.8240	0.8331	0.8476	0.8542

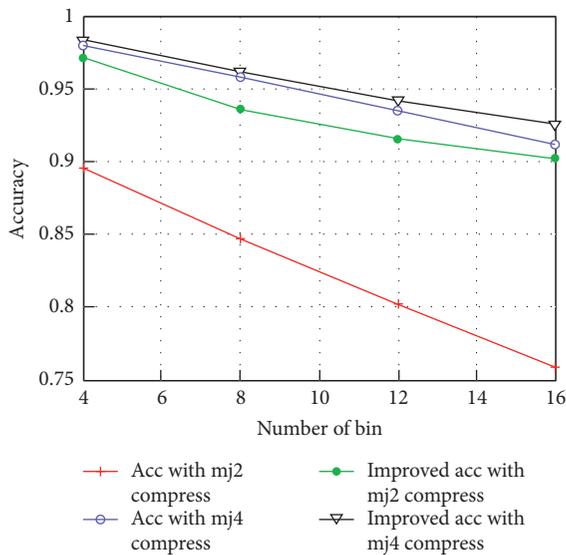


FIGURE 9: Accuracy improvement with compression.

method has good performance with salt and pepper noise and video compression, while it is more sensitive to Gaussian noise compared with other three methods. Although the overall accuracy performance of the proposed method is slightly worse than the DWT method and the DCT method, the accuracy rate still can arrive at 0.97 for most cases. However, the hiding information capacity of our method is 16 bits for every frame, which is twice that of the DWT method and the DCT method during the tests. Therefore, our proposed method has obtained a good trade-off between steganographic capacity and robustness.

**4.5. Hiding Success Rate Analysis.** For coverless information hiding based on feature mapping, the extracted feature sequences should not only ensure the robustness but also reflect the differences of features. Therefore, under the premise of a given database, the success rate of information hiding is also an important indicator to measure the feasibility and the practicability of the secret transmission scheme. Assuming that every hidden information segment

contains  $n$  bits, the number of different mapping sequences that can be generated by the current video library is  $k$ , and the hiding success rate is

$$R_{\text{suc}} = \frac{k}{2^N}. \quad (23)$$

In the test, the videos are chosen randomly from UCF101 and HMDB51 datasets. The success rate of our scheme is compared with the DWT method and the DCT method. The subblock number of all the three algorithms is set as 8. The results are shown in Figure 16. It can be seen that the hiding success rate increases with the increment of the number of the videos. With the same number of videos, the hiding success rate of our method is much higher than other two methods. With only about 70 videos, our method can achieve a hiding success rate as more than 90%. The improvement of hiding success rate comes from the consideration of the optical flow features between adjacent frames in our method. However, the DWT method and the DCT method based on images only focus on every separate frame and the adjacent frames usually have similar texture features. Therefore, the generated bit sequences also have high similarity, which will reduce the hiding success rate.

**4.6. Complexity and Efficiency Analysis.** The complexity of the proposed method mainly lies in the construction of the video index database because the RHOOF of every video frame needs to be calculated. However, the video index database only needs to be constructed once in advance at the sender. During real secret transmission, we only need to consider the time cost of the specific secret information hiding and extraction, in which the main work load lies in the feature analysis of the cover frames. We investigate the efficiency of different methods based on the time cost of hiding information bits with the same length. In Table 5, the time cost of different methods is listed, where “s/B” means the number of seconds that are required for hiding one information byte. It can be seen that the time cost of our method is more than other methods due to the complexity of hierarchical optical flow calculation.

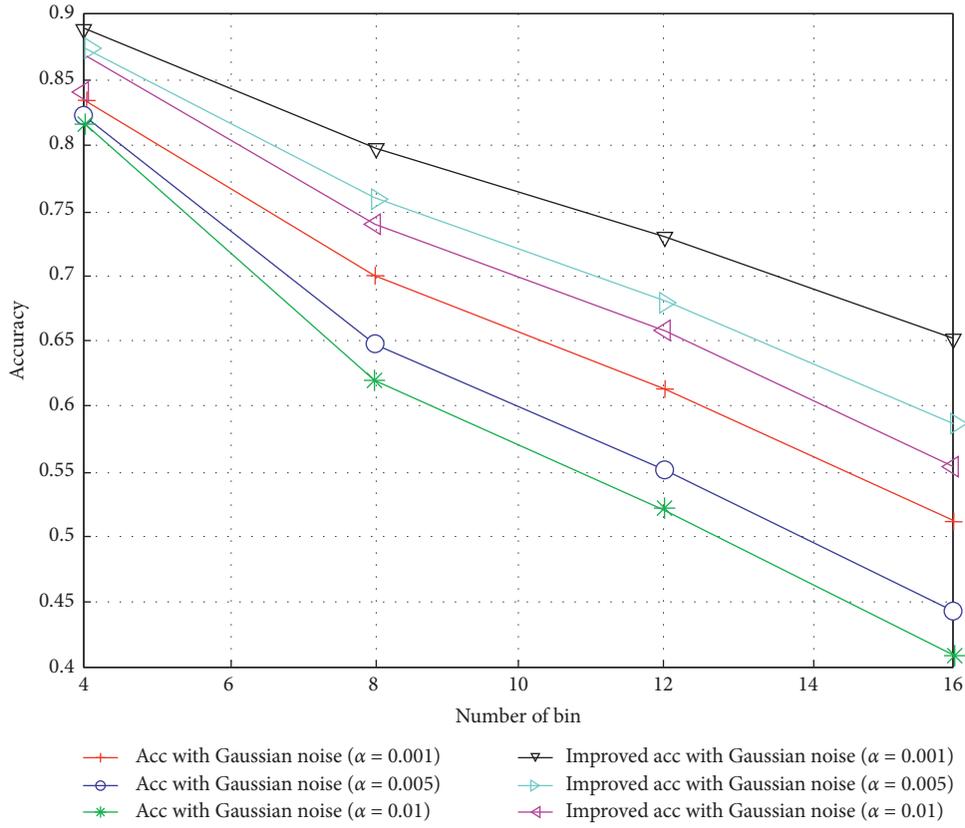


FIGURE 10: Accuracy improvement with Gaussian noise.

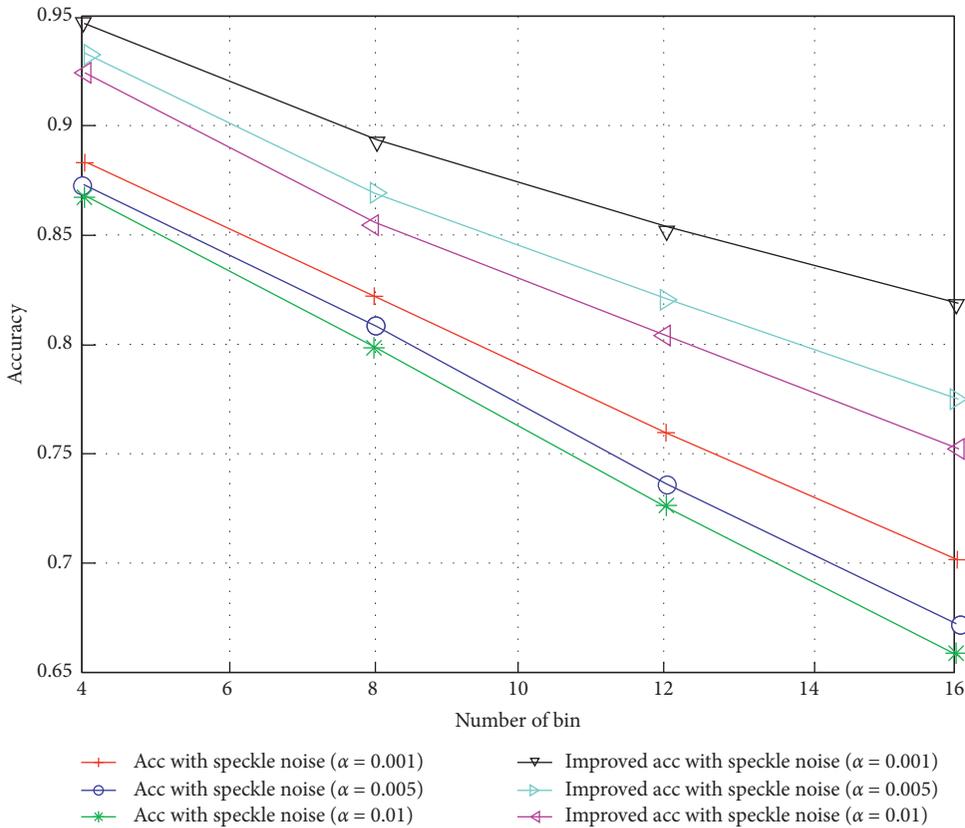


FIGURE 11: Accuracy improvement with speckle noise.

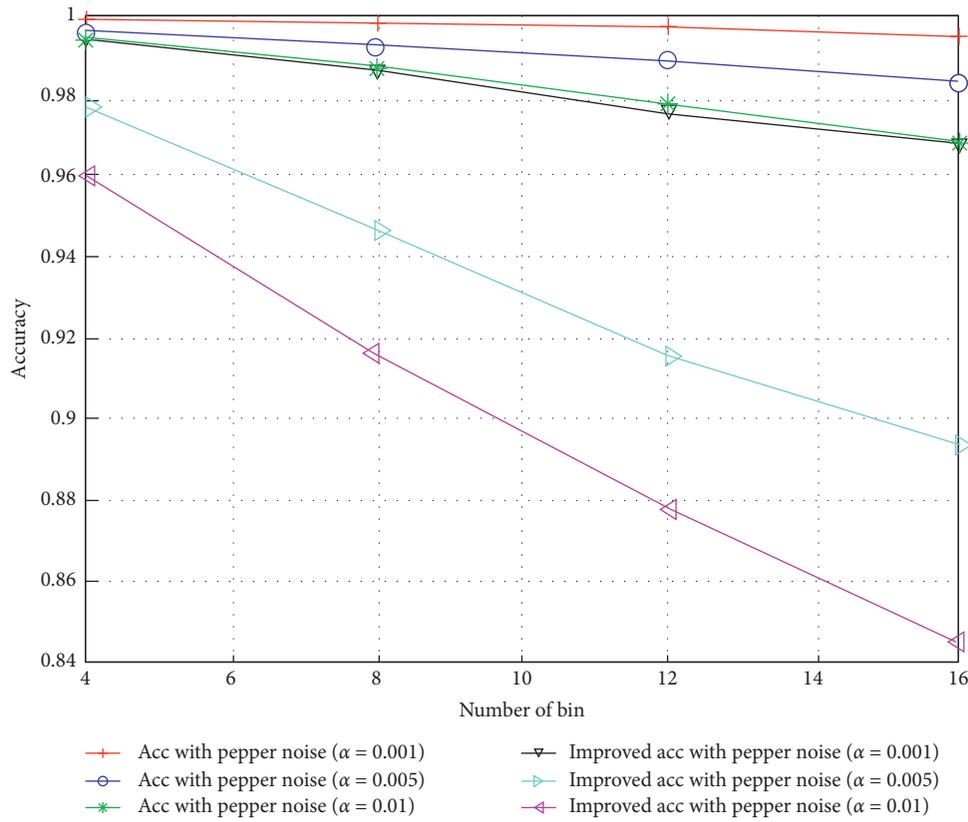


FIGURE 12: Accuracy improvement with salt and pepper noise.

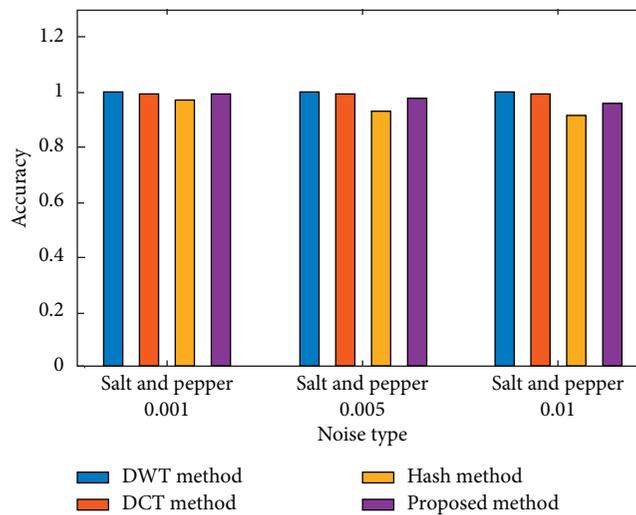


FIGURE 13: Accuracy comparison with salt and pepper noise.

4.7. *Transmission Data Load Analysis.* In the proposed steganography scheme, the video database is shared by both the sender and the receiver. During the information hiding process, the secret information will be preprocessed and mapped to the hash sequences of RHOOF. After searching in the video index database, the corresponding mapping indexes will be sent to the receiver. Then, the receiver can find the cover video from the video database. Therefore, the data

transmission load only includes the contents of the index item as video ID, frame ID, and subblock ID. Assuming the size of the secret information is  $n$  byte, the transmission loads of different methods are analysed in Table 6. Here, the subblock number of the DWT method and the DCT method is 8. It can be seen that the transmission load of our scheme is greatly reduced since the cover video need not to be transmitted. But the sender and the receiver are required to

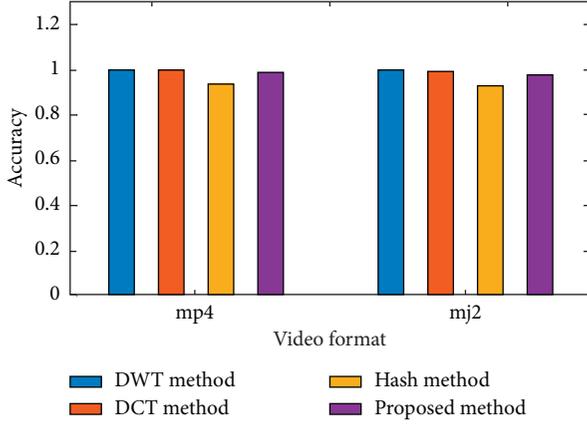


FIGURE 14: Accuracy comparison with video compression.

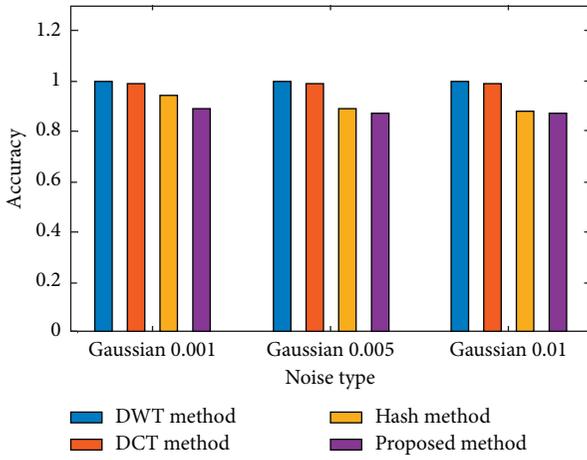


FIGURE 15: Accuracy comparison with Gaussian noise.

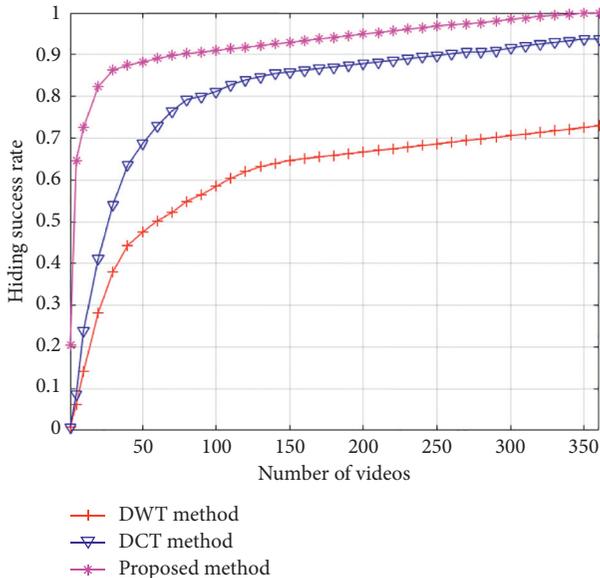


FIGURE 16: Hiding success rate comparison.

TABLE 5: Time cost comparison.

Methods	Hash [20]	DCT [22]	DWT [23]	Proposed
Time cost	0.5006 s/B	0.3748 s/B	0.5085 s/B	1.1874 s/B

TABLE 6: Comparison of transmission data load.

Transmission data load for $n$ byte secret information	
Pixel method [6]	$n$ cover images
Hash method [20]	$\lceil 8n/18 \rceil + 1$ cover images
DCT method [22]	$N + 1$ cover images + side information
DWT method [23]	$n + 1$ cover images + auxiliary information
Proposed method	$n$ indexes

share and update the video database synchronously to ensure the successful information hiding and extraction.

## 5. Conclusion

In this study, a coverless steganography scheme based on motion analysis of videos is proposed. The capacity, robustness, hiding success rate, time cost, and transmission data load have been investigated and compared with the existing methods. It is shown that the proposed method not only obtains a good trade-off between hiding information capacity and robustness but also achieves higher hiding success rate and lower transmission data load, which shows good practicability and feasibility. However, the time cost of our method is higher due to the complexity of hierarchical optical flow calculation. We will try to improve the efficiency in the future work.

## Data Availability

The UCF101 data used to support the findings of this study are available at <http://csrcv.ucf.edu/data/UCF101/UCF101.rar> and the HMDB51 data are available at <http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#Downloads>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61772561 and 62002392), the Key Research and Development Plan of Hunan Province (2019SK2022), the Science Research Projects of Hunan Provincial Education Department (18A174 and 19B584), the Degree and Postgraduate Education Reform Project of Hunan Province (2019JGYB154), the National Natural Science Foundation of Hunan (2020JJ4140 and 2020JJ4141),

and the Postgraduate Excellent Teaching Team Project of Hunan Province ([2019] 370–133).

## References

- [1] B. Wang, W. Kong, W. Li, N. Xiong et al., “A dual-chaining watermark scheme for data integrity protection in Internet of Things,” *Computers, Materials & Continua*, vol. 58, no. 3, pp. 679–695, 2019.
- [2] W. Wan, J. Wang, J. Li et al., “Pattern complexity-based JND estimation for quantization watermarking,” *Pattern Recognition Letters*, vol. 130, no. 1, pp. 157–164, 2020.
- [3] J. Wang, W. B. Wan, X. X. Li, J. D. Sun, and H. X. Zhang, “Color image watermarking based on orientation diversity and color complexity,” *Expert Systems with Applications*, vol. 140, Article ID 112868, 2020.
- [4] Y. Tan, J. Qin, X. Xiang, W. Ma, W. Pan, and N. N. Xiong, “A robust watermarking scheme in YCbCr color space based on channel coding,” *IEEE Access*, vol. 7, no. 1, pp. 25026–25036, 2019.
- [5] B. Chen, Y. Wu, G. Coatrieux, X. Chen, and Y. Zheng, “JSNet: a simulation network of JPEG lossy compression and restoration for robust image watermarking against JPEG attack,” *Computer Vision and Image Understanding*, vol. 197–198, no. 1, pp. 103015–103021, 2020.
- [6] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, “Coverless image steganography without embedding,” *Cloud Computing and Security*, vol. 9483, no. 1, pp. 123–132, 2015.
- [7] J. Qin, Y. Luo, X. Xiang, Y. Tan, and H. Huang, “Coverless image steganography: a survey,” *IEEE Access*, vol. 7, no. 1, pp. 171372–171394, 2019.
- [8] Y. Tong, Y. L. Liu, Y. Liu, J. Wang, and G. Xin, “Text steganography on RNN-generated lyrics,” *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5451–5463, 2019.
- [9] Z.-L. Yang, S.-Y. Zhang, Y.-T. Hu, Z.-W. Hu, and Y.-F. Huang, “VAE-stega: linguistic steganography based on variational auto-encoder,” *IEEE Transactions on Information Forensics and Security*, vol. 16, no. 1, pp. 880–895, 2021.
- [10] L. Y. Xiang, S. H. Yang, Y. H. Liu et al., “Novel linguistic steganography based on character-level text generation,” *Mathematics*, vol. 8, no. 1, pp. 1558–1576, 2020.
- [11] X. Chen, H. Sun, Y. Tobe, Z. Zhou, and X. Sun, “Coverless information hiding method based on the Chinese mathematical expression,” *Cloud Computing and Security*, vol. 9483, pp. 133–143, 2015.
- [12] J. Zhang, J. Shen, L. Wang, and H. Lin, “Coverless text information hiding method based on the word rank map,” *Cloud Computing and Security*, vol. 10039, pp. 145–155, 2016.
- [13] J. Zhang, H. Huang, L. Wang et al., “Coverless text information hiding method using the frequent words hash,” *International Journal of Network Security*, vol. 19, no. 6, pp. 1016–1023, 2017.
- [14] Z. Zhou, J. Qin, X. Xiang, Y. Tan, Q. Liu, and N. Xiong, “News text topic clustering optimized method based on TF-IDF algorithm on spark,” *Computers, Materials & Continua*, vol. 62, no. 1, pp. 217–231, 2020.
- [15] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, “Coverless real-time image information hiding based on image block matching and Dense Convolutional Network,” *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2019.
- [16] J. Qin, H. Li, X. Xiang et al., “An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing,” *IEEE Access*, vol. 7, no. 1, pp. 24626–24633, 2019.
- [17] B. Chen, W. Tan, G. Coatrieux, Y. Zheng, and Y. Q. Shi, “A serial image copy-move forgery localization scheme with source/target distinguishment,” *IEEE Transactions on Multimedia*, vol. 9, pp. 1–12, 2020.
- [18] H. Li, J. Qin, X. Xiang, L. Pan, W. Ma, and N. N. Xiong, “An efficient image matching algorithm based on adaptive threshold and RANSAC,” *IEEE Access*, vol. 6, no. 1, pp. 66963–66971, 2018.
- [19] T. Zhou, B. Xiao, Z. Cai, and M. Xu, “A utility model for photo selection in mobile crowdsensing,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 48–62, 2021.
- [20] S. Zheng, L. Wang, B. Ling, and D. Hu, “Coverless information hiding based on robust image hashing,” *Intelligent Computing Methodologies*, vol. 10363, pp. 536–547, 2017.
- [21] Z. Zhou, Q. M. J. Wu, and C. N. Yang, “Coverless image steganography using histograms of oriented gradients-based hashing algorithm,” *Journal of Internet Technology*, vol. 18, no. 5, pp. 1177–1184, 2017.
- [22] X. Zhang, F. Peng, and M. Long, “Robust coverless image steganography based on DCT and LDA topic classification,” *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3223–3238, 2018.
- [23] Q. Liu, X. Xiang, J. Qin, Y. Tan, J. Tan, and Y. Luo, “Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping,” *Knowledge-Based Systems*, vol. 192, no. 1, pp. 105375–105389, 2020.
- [24] L. Zou, J. Sun, M. Gao et al., “A novel coverless information hiding method based on the average pixel value of the sub-images,” *Multimedia Tools and Applications*, vol. 21, no. 1, pp. 1–16, 2018.
- [25] Y. Cao, Z. Zhou, X. Sun et al., “Coverless information hiding based on the molecular structure images of material,” *Computers, Materials & Continua*, vol. 54, no. 2, pp. 197–207, 2018.
- [26] Y. Tan, J. Qin, H. Tang et al., “Privacy protection for medical images based on DenseNet and coverless steganography,” *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1797–1817, 2021.
- [27] Y. Luo, J. Qin, X. Xiang, and Y. Tan, “Coverless image steganography based on multi-object recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 1–13, 2020.
- [28] T. Xu, M. Zhao, X. Yao, and K. He, “An adjust duty cycle method for optimized congestion avoidance and reducing delay for WSNs,” *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1605–1624, 2020.
- [29] D. Li, L. Qiao, and J. Kim, “A video zero-watermarking algorithm based on LPM,” *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13093–13106, 2016.
- [30] X. Liu, R. Zhao, F. Li, S. Liao, Y. Ding, and B. Zou, “Novel robust zero-watermarking scheme for digital rights management of 3D videos,” *Signal Processing: Image Communication*, vol. 54, no. 1, pp. 140–151, 2017.
- [31] T. Zhang, X. Wang, X. Xu, and C. L. P. Chen, “GCB-Net: graph convolutional broad network and its application in emotion recognition,” *IEEE Transactions on Affective Computing*, vol. 8, pp. 1–12, 2019.
- [32] S. Zhang, C. Gao, F. Chen, S. Luo, and N. Sang, “Group sparse-based mid-level representation for action recognition,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 660–672, 2017.
- [33] S. Cai, Y. Huang, B. Ye, and C. Xu, “Dynamic illumination optical flow computing for sensing multiple mobile robots

- from a drone,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 8, pp. 1370–1382, 2018.
- [34] R. Ke, Z. Li, J. Tang et al., “Real-time traffic flow parameter estimation from UAV video based on ensemble classifier and optical flow,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 54–64, 2018.
- [35] H. Deng, U. Arif, K. Yang, Z. Xi, Q. Quan, and K.-Y. Cai, “Global optical flow-based estimation of velocity for multi-copters using monocular vision in GPS-denied environments,” *Optik*, vol. 219, no. 1, Article ID 164923, 2020.
- [36] M. Sun, W. Sun, X. Zhang, Z. Zhu, and M. Li, “Moving vehicle detection based on optical flow method and shadow removal,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 322, pp. 447–453, 2020.
- [37] C. Lv, Y. Wu, D. Fan, and X. Lu, “Fast registration of UAV aerial images based on improved optical-flow model combined with feature-point matching,” *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8875–8887, 2019.
- [38] J. Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm*, Intel Corporation, Microprocessor Research Labs, Hillsboro, OR, USA, 2000.
- [39] R. Chaudhry, A. Ravichandran, G. Hager et al., “Histograms of oriented optical flow and Binet-Cauchy kernels on non-linear dynamical systems for the recognition of human actions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Miami, FL, USA, 2009.