applied system innovation

MDPI

Article

Exploiting Spatiotemporal Redundancy Using Octree Decomposition to Enhance the Performance of Video Steganography †

Mohammed Baziyad [1,*], Tamer Rabie [2] and Ibrahim Kamel [2]

[1] Research Institue of Sciences & Engineering , University of Sharjah, Sharjah P.O. Box. 27272, United Arab Emirates
[2] Computer Engineering Department, University of Sharjah, Sharjah P.O. Box. 27272, United Arab Emirates; trabie@sharjah.ac.ae (T.R.); kamel@sharjah.ac.ae (I.K.)
* Correspondence: mbaziyad@sharjah.ac.ae

† This paper is an expanded version of Baziyad, M., Rabie, T., & Kamel, I. (2021, October). Utilizing spatiotemporal redundancy to maximize the performance of video steganography: An Octree approach. In Proceedings of the 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 31 October–2 November 2021; pp. 1–6.

Abstract: Leveraging data redundancy has long been recognized as an effective approach for concealing large amounts of secret data. In digital images, the 2D-pixel matrix inherently provides opportunities for redundancy, as each pixel is connected to its eight neighbors. Video segments, with their 3D structures, introduce an additional layer of redundancy known as temporal redundancy. Recent video steganography techniques have proposed utilizing this temporal redundancy for data concealment. This paper seeks to fully exploit the redundancy present in video segments by integrating both spatial and temporal redundancy through an Octree segmentation method. The video is divided into homogeneous, variable-sized 3D cubes to enhance redundancy in each region, thereby improving energy compaction in the 3D discrete cosine transform (3D-DCT) domain. Consequently, the hiding capacity is optimized because most of the signal's energy is concentrated in a few significant 3D-DCT coefficients, leaving a substantial portion of insignificant coefficients. These insignificant coefficients can be replaced with secret data without significantly affecting the quality of the carrier signal.

Keywords: video steganography; DCT; redundancy; capacity; PSNR; bpp

check for updates

Academic Editor: Andrzej Białas

Received: 26 September 2024
Revised: 15 December 2024
Accepted: 20 December 2024
Published: 26 December 2024

Citation: Baziyad, M.; Rabie, T.; Kamel, I. Exploiting Spatiotemporal Redundancy Using Octree Decomposition to Enhance the Performance of Video Steganography. *Appl. Syst. Innov.* **2025**, *8*, 2. https://doi.org/10.3390/asi8010002

Copyright: © 2024 by the authors. Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Steganography is the science that deals with providing confidentiality using obscurity. The secret data are sent to the receiver by hiding their existence within another "innocent" host signal. The output at the sender side is a semi-identical signal to the original host signal and contains the secret data. This signal is named as the carrier signal. Steganography is popular and is preferable by some developers over encryption techniques for the obscurity nature of the communication [1]. Encryption techniques operate by transforming the secret data into an inapprehensible form. This form creates suspiciousness and an eavesdropper can expect the secrecy of the communication. However, a steganography technique does not keep a trace since a secret message is hidden in an "innocent" host signal, which does not trigger the attention of an eavesdropper. With the current advancements in the processing speed of shared networks such as the internet, official entities are using such public networks to send confidential data. Therefore, the need to secure confidential messages over public networks has become a top research area in the last few years [2–5].

Hiding capacity, imperceptibility, robustness, and security are the four main attributes of steganography. Hiding capacity is the amount of hidden secret data compared to the size of the host signal. Imperceptibility refers to the degree of similarity between the host and carrier signal. The robustness aspect refers to the ability of the secret data to withstand attacks. The fourth attribute, the security attribute, deals with the obscurity level and the ability to prevent an eavesdropper from discovering the secrecy of the communication. Developers aim to develop steganography techniques that improve some or all of the four attributes. However, improving some attributes simultaneously is not a trivial process as some attributes such as the hiding capacity and the imperceptibility attributes are opposing each other. The attempt to increase the amount of hidden data will naturally affect the quality of the carrier signal. That is true since hiding is usually performed by replacing some of the original data of the host signal with the secret bits. Therefore, the more data to be hidden, the more original data to be destroyed, which results in higher distortion in the carrier signal.

The capacity–quality trade-off is inherently a limitation in steganography, as the ideal scenario—achieving infinite hiding capacity with zero quality degradation—is not possible in practice. This trade-off reflects a negative phenomenon, as increasing the amount of hidden data typically results in more significant degradation of the stego signal's quality. While efforts can be made to minimize the impact on quality, the trade-off remains a fundamental challenge in balancing the goals of maximizing capacity and preserving the integrity of the carrier signal [6].

For images, low hiding capacity simply means sending low-resolution secret images which can be unacceptable for critical confidential communications. Therefore, it is important to find a solution to increase the hiding capacity without compromising the quality of the carrier image. A common approach to hiding confidential data involves replacing the less significant bits of the host signal with the secret bits. By adopting such an approach, it is possible to hide a significant amount of secret data while maintaining the quality of the carrier signal. The challenging issue is the process of figuring out these less significant bits and whether destroying these bits would significantly damage the quality of the carrier signal. There are two approaches to finding the less significant bits. The first approach is biological and based on the abilities of the human visual system. It is known that the human visual system is less sensitive to noise distributed in high-frequency areas of an image such as edges. Therefore, a popular steganography approach is to hide the secret data in the edge areas since any distortions that occur because of the embedding process will not be visually clear. The other approach is statistically based. In this approach, the embedder captures the available redundant data in the host signal and then replaces the redundant data with the secret bits.

A digital video signal contains two sources of redundancy. The spatial relations between pixels are a source of redundancy. The other source of redundancy is the temporal relation between pixels. The amount of redundant data is related to several attributes such as the motion speed of the objects and the amount of details in each still frame. A video segment that has frames with little details and slow-motion content will have a significant amount of redundant data.

Exploiting data redundancy is a key strategy for compression and steganography. The redundant data in a signal can be captured by analyzing the frequency content of the signal. This is done by transforming the signal from the native domain to the frequency domain using a transform technique. The discrete cosine transform (DCT) is a successful transform technique that is widely favored by data-hiding techniques. Similar to the discrete Fourier transform (DFT), the DCT breaks the input signal into its fundamental frequency components. The critical advantage of the DCT over DFT is the strong compaction property

of the DCT. For steganography, the DCT has a strong ability to express a signal within a few critical frequency components. The strong compaction property results in having a respectful amount of insignificant frequency components, which can be replaced with the secret bits without affecting the quality of the carrier signal.

It has been noticed that the strong compaction property of the DCT becomes stronger with a highly correlated dataset [7]. This means that fewer significant frequency components can hold most of the signal's energy compared to an uncorrelated dataset. Therefore, several recent steganography schemes suggested segmenting the host signal to correlated segments to fully exploit the strong compaction property of the DCT. For example, a recent video steganography scheme has been proposed to extract a temporal vector from each spatial location [8]. The rationale of the proposed technique is that consecutive frames will surely have correlated pixels to develop a valid video segment. This segmentation process has ensured capturing the temporal redundancy to some extent.

The work proposed in [8] does not optimally capture the redundant data in a video segment. Therefore, this paper proposes a new video steganography technique that aims to integrate both the temporal and the spatial redundancy by applying the Octree segmentation technique which is a spatiotemporal segmentation algorithm. The Octree segmentation technique can break up a 3D signal into smaller homogeneous 3D signals. The proposed scheme suggests applying the 3D-DCT on each homogeneous 3D signal. The energy of each homogeneous 3D signal will be packed within a few numbers of significant frequency components. Therefore, a respectful amount of insignificant frequency components can be replaced safely with a scaled version of the secret data. Thus, the proposed steganography technique can deal efficiently with the capacity–imperceptibility trade-off problem and hide a substantial number of secret bits while maintaining the quality of the carrier signal.

The rest of this paper is organized as follows. Section 2 presents related works in the video steganography field. In Section 3, the proposed Octree steganography technique is described. Comparative experimental analysis is presented in Section 4. Section 5 discusses future plans. Finally, our concluding remarks appear in Section 6.

2. Related Works

Video steganography systems can be classified into three main classes, namely still picture steganography, metadata methods, and raw video techniques [9,10]. The still picture steganography views video steganography as an extension of image steganography. Since a video segment is composed of sequential still images, still picture steganography methods are based on embedding the confidential data frame by frame independently. Still picture steganography can be subdivided into native domain technique, and frequency domain techniques. The least-significant-bit (LSB) techniques are the most classical approaches in steganography. The LSB technique hides the secret data in the least significant bit planes of the frames of the video segment. LSB techniques are favorable due to their simplicity and low computational complexity. However, it has been widely known that data hidden using simple LSB substitution techniques can be easily discovered using basic steganalysis methods [11]. Therefore, recent LSB techniques are usually integrated with other techniques to strengthen the steganography system. A video LSB-based technique is proposed in [12]. The proposed technique aimed to improve the LSB performance by integrating the basic LSB technique with the genetic algorithm (GA). This proposed LSB-GA technique was able to achieve an enhanced quality of the carrier video signal when hiding at a similar rate to basic LSB techniques. The carrier quality reached up to 39.37 dB while hiding with an average of 2.66 bits per gray-scale pixel. Another video LSB-based method is presented in [13]. This proposed method used only one LSB plane for each frame from the host segment for hiding. As a result, this technique was able to achieve a high

imperceptibility level since only one LSB is utilized for hiding. However, the embedding rate is low compared to other steganography systems in the literature.

The tri-way pixel-value differencing (TPVD) is another spatial domain technique [14]. The tri-way pixel-value differencing (TPVD) technique is based on the popular pixel-value differencing (PVD) technique. The PVD technique hides the confidential data in the difference value of two adjacent pixels. A TPVD-based video steganography technique is proposed in [15]. The proposed technique adopted the TPVD algorithm to hide in a compressed domain of the host video segment.

The other subclass under the still pictures method is the frequency-domain method. These techniques convert the host frames from their native spatial domain into a domain that facilitates frequency analysis. The transform can be done using one of the frequency analysis tools such as the discrete Fourier transform (DFT), discrete cosine transform (DCT), and discrete wavelet transform (DWT). The confidential data are then hidden by replacing some selected frequency coefficients with the secret data. A DWT-based technique is presented in [16]. This proposed technique used the Bose–Chaudhuri–Hocquenghem (BCH) codes to encode the secret bits and then hide them in the DWT coefficients of the video frames. The proposed technique embedded the confidential data in all DWT coefficients excluding the low-frequency DWT coefficients. The reason is that low-frequency coefficients contain the most critical parts of the signal, which must remain intact to maintain an acceptable quality of the carrier frame. The proposed technique achieved a higher hiding rate compared to similar approaches in the literature. Another DWT scheme was proposed in [17]. This DWT technique is a blind scheme where embedding is performed in human skin regions in a frame. Hiding is done in the blue and red channels using a wavelet quantization method. The proposed scheme was tested against MPEG-4 compression and showed a good robustness level.

The second class is the metadata schemes. These schemes target a specific video format by exploiting the structure and the compression algorithm adopted in the format. An example of a popular standard video format is the H.264/AVC format. This standard has a high ability to compress a video segment with a high compression ratio. Thus, it has been widely used in network transmissions. In [18], a new data-hiding scheme based on H.264 was presented. In the proposed scheme, confidential data is hidden by modifying the sign flag of trailing ones and the levels' words in the context-adaptive variable length coding (CAVLC), which is used in the entropy coding of the H.264 technique.

A different method based on H.264 is introduced in [19]. This approach conceals secret data within the high-frequency coefficients (trailing ones) of CAVLC coding. Its main advantage lies in its low computational complexity compared to other H.264/AVC-based hiding techniques, making it suitable for real-time implementation. However, this method is not designed for high hiding capacity, as it only embeds 250 bits in a 352×288 video with 30 frames.

The third category consists of raw video methods. These methods aim to leverage the 3D nature of videos by using the time dimension t for embedding. This extra dimension provides additional properties like motion vectors and motion components [9]. In [20], an adaptive steganography scheme embeds data in a compressed video stream by utilizing the temporal and spatial features of the host video along with the concept of the human visual system.

A different hiding scheme utilizing discrete wavelet transform (DWT) and discrete cosine transform (DCT) is introduced in [21]. This method employs a multiple object tracking (MOT) algorithm and error-correcting codes for the embedding process. Initially, the "Hamming and Bose" algorithm and the "Chaudhuri and Hocquenghem" algorithm are used to encode the secret data. Next, the motion-based MOT algorithm is applied to

the host video to identify the areas of interest in moving objects. Finally, the secret message is embedded into the DWT and DCT coefficients of all motion regions in the video, based on the foreground mask. This scheme can embed a secret message that occupies 3.40% of the host video, with a carrier quality of 49.0 dB.

In [22], a robust video watermarking method is introduced to embed a watermark image into digital video frames using the variable-temporal length 3D-DCT technique. This approach involves selecting a series of consecutive 8×8 blocks and applying the 3D-DCT to these blocks. The embedding process occurs in the mid-range coefficients of the individual blocks.

This paper introduces a novel video steganography technique that integrates both temporal and spatial redundancy by utilizing the Octree segmentation technique, a spatiotemporal segmentation algorithm. The Octree segmentation technique divides a 3D signal into smaller homogeneous 3D segments. The proposed method applies the 3D-DCT to each homogeneous 3D segment. The energy of each homogeneous 3D segment is concentrated in a few significant frequency components, allowing a considerable amount of insignificant frequency components to be safely replaced with a scaled version of the secret data. As a result, this steganography technique effectively addresses the capacity–imperceptibility trade-off, enabling the hiding of a substantial number of secret bits while preserving the quality of the carrier signal.

3. The Proposed Octree-Hiding Scheme

The Pixogram segmentation technique outlined in Section 2 clearly utilizes temporal redundancy. However, it overlooks another source of redundancy: spatial redundancy. Given that a video segment is essentially a sequence of static images (frames), there is a potential for significant redundant data within each frame. To address this, the Octree segmentation method is proposed to leverage both spatial and temporal redundancies. The Octree segmentation technique is a 3D extension of the quad-tree image segmentation method. While the quad-tree method focuses on the spatial homogeneity of pixel intensities to divide an image into non-overlapping, homogeneous blocks, the Octree method extends this concept to three dimensions. It segments a 3D volume into non-overlapping, homogeneous blocks based on spatial correlations within the pixel intensities.

In [23,24], audio steganography techniques are proposed that share a similar rationale to the technique discussed in this paper. These methods segment the cover signal into correlated segments and exploit the strong compaction property of the discrete cosine transform (DCT) within these segments. However, the segmentation in these audio-based techniques was performed in the 1D domain, whereas the proposed technique in this paper employs a 3D segmentation approach tailored for video steganography. This distinction underscores the progression of segmentation strategies from audio to video, illustrating the versatility of using DCT compaction properties in different domains for robust steganographic embedding.

The quad-tree process begins by dividing the image into four equal squares. Each square is then evaluated to determine if it meets the homogeneity criteria. If the block does not meet the criteria, it is further subdivided into four smaller squares. This process of division is repeated iteratively until all blocks satisfy the homogeneity requirement. A straightforward homogeneity criterion involves checking the difference between the highest and lowest gray-level values within a block. If this difference is below a predefined threshold T_{quad}, the block is considered homogeneous.

The quad-tree algorithm can be configured with both a minimum and maximum block size. This ensures that the algorithm does not create blocks smaller than the minimum specified size (e.g., 8×8) or larger than the maximum specified size (e.g., 128×128). Even if

larger blocks meet the homogeneity threshold, they will be split to adhere to the maximum block size constraint. By the end of the quad-tree process, the image will be segmented into variable-sized blocks that reflect the pixel correlations. Figure 1a illustrates the application of the quad-tree segmentation technique.

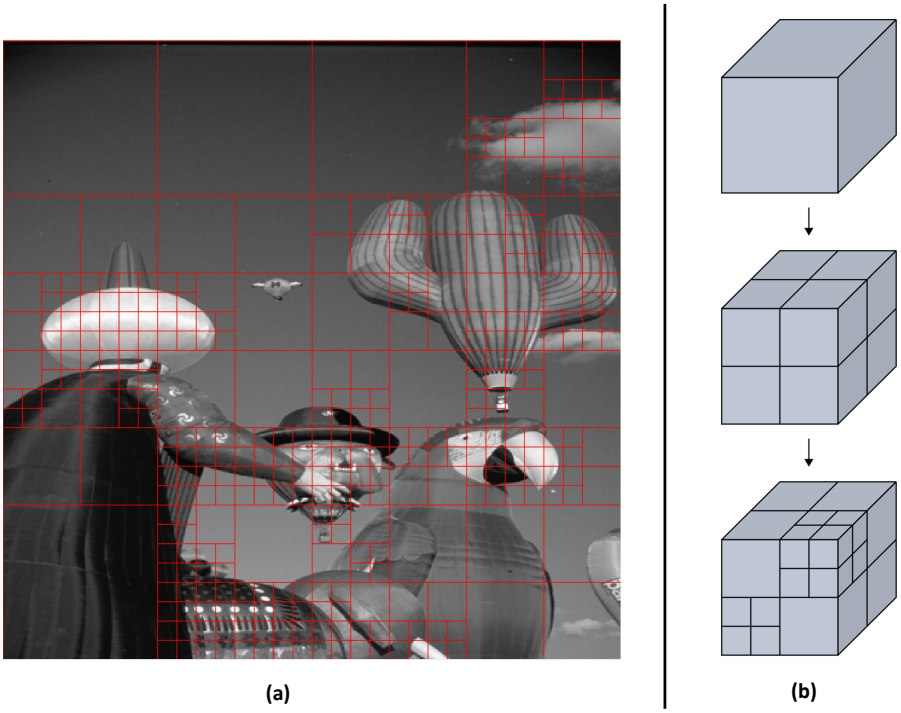


Figure 1. (**a**) Applying the 2D quad-tree segmentation method on an image when setting a minimum block size of 32 × 32. (**b**) The 3D Octree segmentation method.

The Octree method functions in a similar manner. As shown in Figure 1b, the Octree segmentation technique extends the quad-tree approach to 3D signals, such as video segments. Instead of dividing a 2D image into correlated blocks, the Octree segments a 3D video segment into correlated cubes. This segmentation process follows the same principles as the quad-tree technique but operates in three dimensions. A homogeneous cube thus captures both spatial and temporal redundancies.

As depicted in Figure 2, the process begins with selecting a homogeneous block obtained through the Octree segmentation technique. Following this, the 3D discrete cosine transform (3D-DCT) is applied to the homogeneous block. The 3D-DCT is computed using the formula given in [25]:

$$Y_{k_1,k_2,k_3} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \sum_{n_3=0}^{N_3-1} y_{(n_1,n_2,n_3)} \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right] \cos\left[\frac{\pi}{N_3}\left(n_3+\frac{1}{2}\right)k_3\right] \quad (1)$$

where:

N_1 represents the length of the signal (in samples);
N_2 denotes the width of the signal (in samples);
N_3 is the depth of the signal (in samples);
k_1 ranges from 0 to $N_1 - 1$;
k_2 ranges from 0 to $N_2 - 1$;
k_3 ranges from 0 to $N_3 - 1$;
Y_{k_1,k_2,k_3} denotes the DCT coefficient at (k_1, k_2, k_3);

y_{n_1,n_2,n_3} is the time-domain sample at (n_1, n_2, n_3).

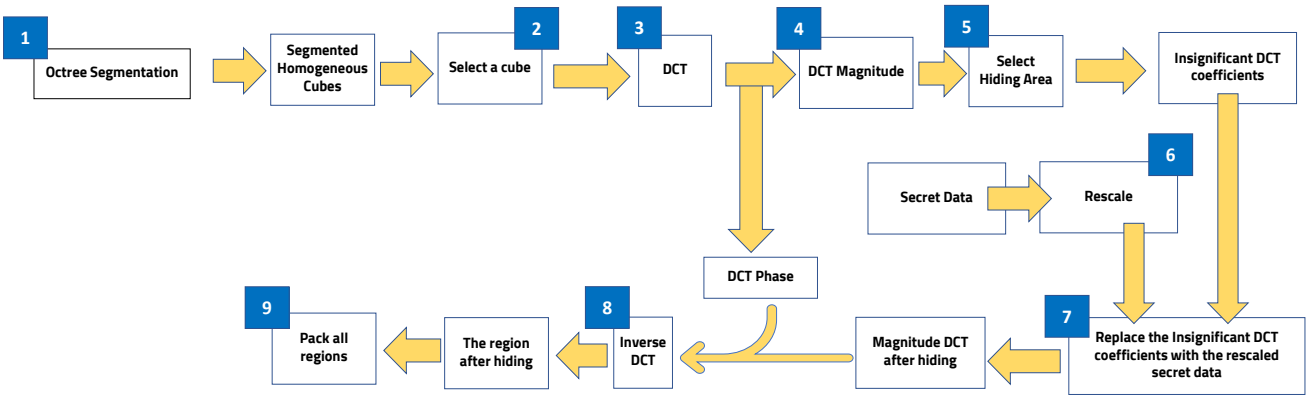


Figure 2. General schematic representation of the proposed Octree steganography technique.

Recent research has demonstrated that the magnitude of the DCT coefficients is less critical to the signal than the phase [26]. Consequently, the proposed method focuses on separating the magnitude from the phase, hiding information solely in the DCT magnitude of the host image, as illustrated in step 3 in Figure 2.

Next, the quantization process begins to identify the insignificant DCT coefficients that will be used to embed the secret data. Figure 3 illustrates the distribution of significant DCT coefficients within a 3D-DCT matrix [27]. To model the shaded volume depicted in Figure 3, we first consider the $x - y$ plane where $t = 0$. The shaded area in the $x - y$ plane can be approximated as a hyperbola, such that we have the following:

$$xy \leq C_1 \tag{2}$$

Similarly, the shaded areas in the $x - t$ and $y - t$ planes can be represented, respectively, by the following:

$$xt \leq C_2 \tag{3}$$

$$yt \leq C_3 \tag{4}$$

where C_1, C_2, and C_3 are constants. We combine the following equations: (2)–(4):

$$(xyt)^2 \leq C_1 C_2 C_3 \tag{5}$$

Thus, the important coefficients $g(x, y, t)$ can be allocated by the following:

$$g(x, y, t) = xyt \leq \sqrt{C_1 C_2 C_3} \tag{6}$$

Since C_1, C_2, and C_3 are constants, the significant coefficients $g(x, y, t)$ can be founded by a single constant C':

$$g(x, y, t) = xyt \leq C' \tag{7}$$

The embedding process is carried out within the insignificant DCT coefficients of each cube. Given that these coefficients have relatively low values, the system, as depicted in step 6 in Figure 2, adjusts the amplitude of the secret signal by applying a scaling factor k. This scaling aligns the secret signal's amplitude with the range of the insignificant DCT coefficients. This adjustment can be described by the following equation:

$$\text{Rescaled Signal} = k \cdot \text{Secret Signal} \tag{8}$$

where k is the scaling factor used to modify the amplitude of the secret signal so that it fits within the range of the insignificant DCT coefficients.

$$D' = \frac{1}{k} * S \tag{9}$$

where D' is the secret vector after rescaling and S denote the original secret data.

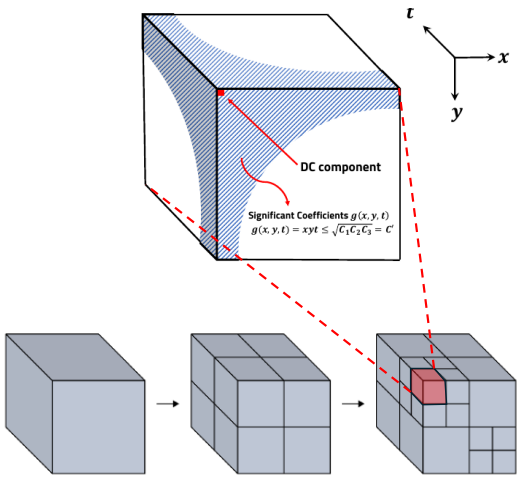


Figure 3. Locating the area of the significant coefficients in each cube using the following equation: $xyt \leq \sqrt{C_1C_2C_3} = C'$.

Once the secret signal is rescaled using an appropriate scaling factor, the next step involves substituting the insignificant DCT coefficients with the rescaled secret signal (as shown in step 7 in Figure 2). Since the embedding is carried out only in the magnitude of the DCT coefficients, the original phase spectrum must be preserved and combined with the altered magnitude DCT to fully reconstruct the DCT structure, ensuring that the secret data is embedded correctly.

The next step involves applying the inverse DCT to convert the cubes back into the spatial domain. It is important to note that these embedding procedures are performed on each cube segmented by the Octree technique. Once the secret data are embedded in all cubes and their spatial representations are obtained, the embedder assembles all the cubes back into their original positions within the video segment, resulting in the final carrier video that incorporates the hidden data.

3.1. The Extraction Process

Three different methodologies are employed to optimize the quality and efficiency of secret data recovery: semi-blind extraction, non-blind extraction without error correction, and non-blind extraction with error correction. The semi-blind approach allows the receiver to extract hidden information by utilizing minimal parameters shared by the sender, resulting in reduced communication costs, albeit at the risk of extraction quality. In contrast, the non-blind extraction method demands comprehensive parameter transmission, ensuring higher extraction fidelity but potentially leading to increased communication overhead. Lastly, the non-blind approach with error correction seeks to balance extraction quality and communication efficiency by transmitting only error correction data, thus refining the output while minimizing the amount of information exchanged. Each of these techniques offers unique advantages and challenges, shaping the effectiveness of the data extraction process.

3.1.1. Semi-Blind Extraction Approach

The receiver should be able to extract the secret data following the reverse order of the procedures described earlier on the sender side. Figure 4 shows a general illustration of the secret extraction process.

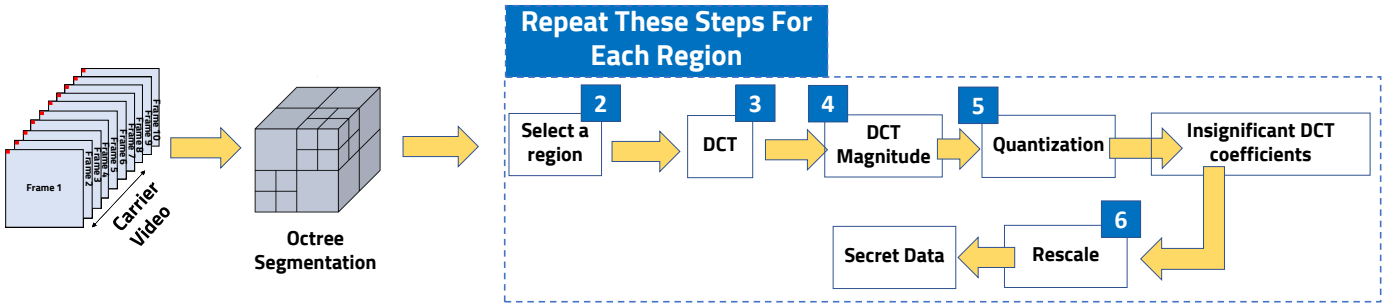


Figure 4. A general illustration of the extraction process.

Once the carrier video is received, the homogeneous regions are extracted using the same segmentation technique used at the embedder. Next, the quantization process is performed on the DCT vector to locate the indices of the insignificant DCT coefficients that contain the secret data. Finally, the secret data are re-scaled back to the natural original range.

The previously described method of extraction is the semi-blind extraction version of the proposed hiding system. The advantage of the semi-blind approach extraction is the low communication cost since the sender will need to send a few parameters. Table 1 lists the parameters that need to be sent to the receiver to be able to extract the hidden data. This semi-blind way of extraction can work if the carrier video has a very high quality and is close to being identical to the host video. In this case, the extraction algorithm will be able to generate the same exact regions generated at the embedder, and thus, the secret data will be extracted in acceptable quality.

Based on Table 1, the receiver must receive the threshold parameter T_{quad} to be able to segment the regions of the carrier signal using the same threshold value. Moreover, the scaling factor k is needed to be able to reconstruct the original secret message. Moreover, the proposed Octree segmentation method needs two additional parameters. The first parameter is the minimum block size to be able to obtain the same Octree cubes obtained at the sender. The second parameter is the C′ parameter to be able to locate the insignificant coefficients in each cube.

Table 1. The parameters that need to be sent to the receiver to be able to extract the hidden data.

Method	Parameters	Quantity
Semi-Blind	Min. block size	One value for the entire video segment
	Parameter C′	
	Scaling Factor k	
No Error Correction	The Octree structure	One structure for the entire segment
	Parameter C′	One value for the entire segment
	Scaling Factor k	
Error Correction	The error bit	One value for each error
	Parameter C′	One value for the entire segment
	Scaling Factor k	

3.1.2. Non-Blind Approach—No Error Correction

The semi-blind extraction method might fail to extract the secret data with acceptable quality due to some statistical differences between the host and carrier video segments that can be imperceptible by visual means. To ensure extraction with the best extraction quality, the sender must send all of the parameters that are essential for the extraction process, such as the output of the Octree segmentation. However, this approach will maximize the communication cost, which can be impractical for large-sized video segments. Table 1 lists all of the parameters needed to extract the secret message using the non-blind approach.

The structure information of the Octree can be simplified using the Z-order sorting method. Figure 5 shows the implementation of the Z-order method on a quad-tree. The Z-order sorting method works by analyzing the quad-tree blocks level by level in a Z-order manner (top-left block → top-right block → bottom-left block → bottom-right block). The quad-tree shown in Figure 5 was described efficiently by a tree structure. The tree structure shows three levels, which represent the segmentation levels in the quad-tree. The first level has a value of 1, which means that initially the input image is divided into four non-overlapping blocks. The second level of the tree structure shown in Figure 5 represents the splitting procedure in the second level. The second level had a sequence of [0 1 1 1], which simply means the following commands:

1. Do not split the top-left block.
2. Split the top-right block.
3. Split the bottom-left block.
4. Split the bottom-right block.

The third level is represented in the same way that level 2 was represented. As shown in Figure 5, the third level shows the segmentation procedure on the blocks that were segmented in level 2. Blocks that were not segmented in level 2 will not have a child structure in level 3. At the end, all ones and zeros are packed in one vector in order starting from the first tree level and ending with the last tree level as shown in Figure 5. In the non-blind approach, the sender can only send the resultant vector as the quad-tree structure representative where the receiver can easily generate the quad-tree structure from the sent vector. Similarly, the Octree can be also represented by the same z-order method described in Figure 5. However, instead of having four child nodes for each parent, the Octree structure will have eight child nodes for each parent.

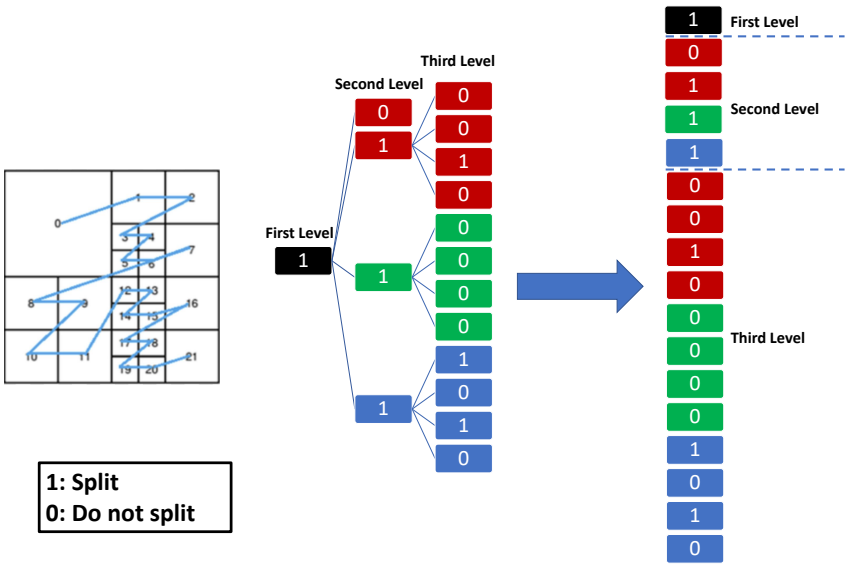


Figure 5. Simplifying the information to be sent of the quad-tree structure using the Z-order method.

3.1.3. Non-Blind with Error Correction Approach

To compensate for the quality of the extracted image and the communication cost, a third extraction methodology is proposed. In this approach, the sender must first obtain the Octree cubes from both the carrier and host video segments. After that, the sender calculates the error between the two region sets and sends only the error information. At the receiver, the cubes are obtained from the carrier video and then combined with the error information received from the sender. Based on Figure 5, the output of the Octree representation is a vector consisting of zeros and ones. If an error has occurred within a specific index, the sender must send the correction of that specific index to the receiver.

4. Comparative Results

This section details the experiments conducted to assess the effectiveness of the proposed Octree-hiding technique. The tests involved embedding data into three distinct video segments with varying motion dynamics. The first video, "Monitoring", is sourced from a live feed captured by a monitoring system. The second, "Documentary", is a clip from a scientific documentary, while the third, "Film", is a segment from a feature film. Figure 6 provides a visual example of embedding the "Baboon" secret image into the "Film" video segment. The figure demonstrates that there is no noticeable visual difference between the original host frames and the carrier frames, with the quality of the extracted secret image being satisfactory.

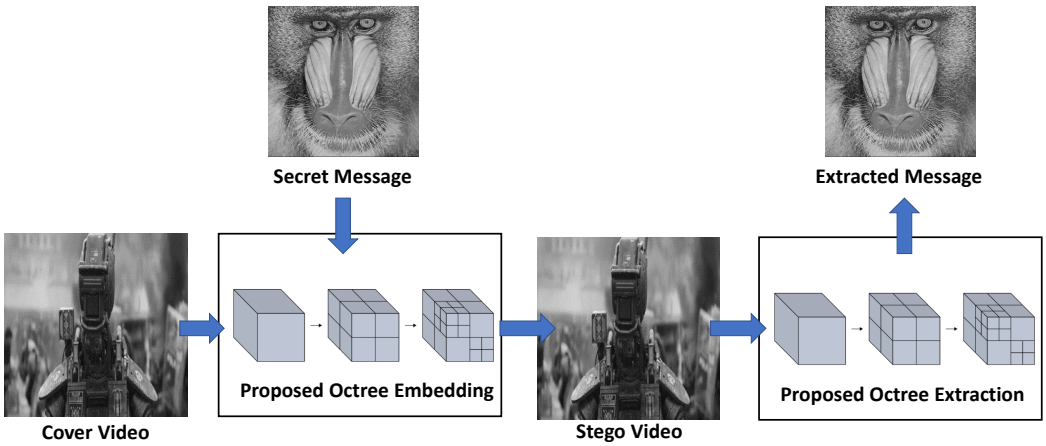


Figure 6. Showing visually the secret, host, carrier, and the extracted images when hiding using the proposed Octree technique.

Table 2 displays the results of data embedding in both fast-motion and slow-motion video segments from the three selected host video sources. It is evident from Table 2 that the performance of the proposed Octree technique varies considerably depending on the motion characteristics of the video segment. The Octree method performs exceptionally well with slow-motion video segments. As previously discussed, the Octree technique is designed to leverage both spatial and temporal redundancies in video segments. Slow-motion video segments inherently contain a significant amount of temporal redundancy, providing an advantage for the proposed method.

Table 3 provides a comparison between the proposed Octree steganography method and recent data-hiding techniques, evaluating both the embedding ratio and carrier quality. The embedding ratio is assessed using the bit-per-pixel (bpp) metric, while carrier quality is measured with the peak-signal-to-noise-ratio (PSNR) metric. Further details on bpp and PSNR are available in [28,29].

Table 2. The achieved rates for the embedded capacity and carrier quality metrics obtained by the suggested Octree concealment method for video segments with slow or fast motion.

Motion	Hosting Video	Hiding Rate (/24)	Carrier Quality (dB)
Slow	Monitoring	23.16	28.57
	Documentary	23.16	32.51
	Film	23.16	32.74
Fast	Monitoring	17.65	31.76
	Documentary	17.65	36.37
	Film	17.65	46.95

Table 3. Comparative results are presented as maximum bbp/dB values for the different methods, with the highest capacity values highlighted in bold.

Method	bpp	dB
[19]	2.4×10^{-4}	**43.15**
[17]	0.008	**46**
[5]	7.98	**44.34**
[12]	7.98	34.37
[30]	7.8×10^{-5}	38.33
[18]	1.73	**42.03**
[21]—DWT	0.2720	**48.67**
[21]—DCT	0.2768	**49.01**
[31]	5.256	34.66
[16]	1.2	**45.68**
[8]	**13.96**	25.29
[32]	8.19	22.14
[33]	7.75	41.5
[34]	22.89	30.05
[6]	22.89	30.22
Proposed Octree—slow motion	**23.16**	32.74
Proposed Octree—fast motion	**17.65**	**46.95**

Table 3 presents a comparison between the proposed Octree scheme and other recent steganography methods. It is evident from the table that the proposed Octree scheme surpasses other state-of-the-art techniques in both hiding capacity and carrier quality. Specifically, the Octree scheme outperforms the other video steganography methods listed in terms of payload capacity. Additionally, the carrier quality achieved by the Octree scheme is superior to several competing methods, even with a significantly higher payload capacity. For instance, while the proposed Octree scheme achieves a capacity rate of 17.65 bpp compared to 7.98 bpp of Dasgupta et al. [12], it also delivers better carrier quality, with an imperceptibility of 46.95 dB versus 44.34 dB.

The proposed technique demonstrates superior performance, particularly in slow-motion segments, by effectively leveraging the compaction property of the discrete cosine transform (DCT). The DCT excels in representing a signal using a minimal number of concentrated coefficients, especially in scenarios with high spatial and temporal correlation,

such as slow-motion video. This property allows for efficient energy compaction, concentrating the majority of the signal's energy into a few significant coefficients while leaving a large area of insignificant ones.

The rationale behind the proposed Octree technique is to exploit this property strategically. By targeting the large area of insignificant coefficients generated by the DCT, the Octree method provides a structured and efficient way to hide secret data. This approach not only increases the hiding capacity but also minimizes visual or perceptual degradation, as the hidden data resides in areas that contribute minimally to the overall signal quality.

The hierarchical structure of the Octree further enhances the efficiency of this technique. It allows for precise localization and adaptive utilization of the insignificant coefficients, ensuring that the embedding process is both scalable and robust. This combination of the DCT's compaction ability and the Octree's hierarchical design results in a technique that achieves higher capacity for data-hiding while maintaining lower degradation in the signal, making it ideal for applications requiring both high data concealment and quality preservation.

In host video segments with fast-motion content, temporal redundancy is reduced compared to slow-motion segments, which can lead to diminished performance. However, the proposed Octree scheme demonstrates better resistance to performance degradation in fast-motion scenarios. This is because the Octree scheme leverages both spatial and temporal redundancy. While fast-motion segments may offer less temporal redundancy, the spatial redundancy within individual frames still contributes significantly to the overall redundant information. Consequently, the Octree scheme compensates for the lack of temporal redundancy by utilizing spatial redundancy, thereby effectively hiding larger amounts of secret data.

Tables 4–6, Tables 7–9, and Tables 10–12 show the results obtained with the proposed Octree-hiding scheme when hiding using the three different video segments "Monitoring", "Documentary", and "Film", respectively, with different motion content. Based on Tables 4–6, Tables 7–9, and Tables 10–12, the proposed Octree scheme has achieved outstanding results that reached up to hiding with 23.16 bpp while achieving a carrier quality of 34.64 dB. This result was obtained when hiding the into the "Film" segment while setting the *C* threshold value to 0.95.

Table 4. Results of the proposed Octree technique when using the **Monitoring slow-motion segment** as the host video segment with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Monitoring	6900–6964	0.95	25	23.16	28.15	0.8214	75.82
			50	23.16	28.57	0.8446	76.5
			75	23.16	28.57	0.8456	76.52
			100	23.16	28.53	0.8446	76.49
		0.85	25	20	33.12	0.9332	73.27
			50	20	35.33	0.9655	74.63
			75	20	35.66	0.9702	74.83
			100	20	35.7	0.9712	74.87
		0.75	25	17.65	34.67	0.9492	69.17
			50	17.65	38.47	0.9813	70.92
			75	17.65	39.39	0.9865	71.28
			100	17.65	39.63	0.9879	71.38

Table 5. Detailed results of the proposed Octree-hiding technique using **Monitoring medium-motion segment** as the host video segment with a $32 \times 32 \times 32$ minimum cube size.

Host	Frames	C	Scale k	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Monitoring	1–64	0.95	25	23.16	26.33	0.7934	74.66
			50	23.16	25.95	0.7707	74
			75	23.16	25.76	0.7584	73.64
			100	23.16	25.64	0.7513	73.43
		0.85	25	20	30.79	0.9239	72.46
			50	20	30.44	0.9194	72.26
			75	20	30.13	0.9136	72.04
			100	20	29.95	0.9098	71.9
		0.75	25	17.65	33.11	0.954	68.9
			50	17.65	32.89	0.9542	68.85
			75	17.65	32.51	0.9504	68.66
			100	17.65	32.28	0.9477	68.53

Table 6. Detailed results of the proposed Octree-hiding technique using **Monitoring fast-motion segment** as the host video segment with a $32 \times 32 \times 32$ minimum cube size.

Host	Frames	C	Scale k	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Monitoring	5–69	0.95	25	23.16	24.33	0.7956	74.21
			50	23.16	24.03	0.7838	73.84
			75	23.16	23.88	0.7767	73.63
			100	23.16	23.8	0.7726	73.5
		0.85	25	20	29.18	0.9234	72.04
			50	20	28.84	0.9232	71.95
			75	20	28.6	0.9201	71.81
			100	20	28.45	0.9179	71.72
		0.75	25	17.65	31.76	0.9538	68.56
			50	17.65	31.53	0.9574	68.59
			75	17.65	31.23	0.9556	68.47
			100	17.65	31.05	0.9542	68.39

The detailed results presented in the tables highlight the performance of the proposed Octree-hiding technique across various video segments and motion types. By comparing bit-per-pixel (bpp) values and their corresponding peak signal-to-noise ratio (PSNR) measurements, we gain insights into the trade-offs between data embedding capacity and video quality.

In the "Monitoring" slow-motion segment (Table 4), it is observed that as the capacity (bpp) decreases from 23.16 to 17.65, the PSNR improves significantly. For instance, with $C = 0.75$ and $k = 100$, the PSNR reaches 39.63 dB compared to 28.15 dB at the highest capacity of 23.16 bpp. This trend indicates that lower embedding capacities allow for higher video quality in terms of PSNR.

Table 7. Results of the proposed Octree technique when using the **Documentary slow-motion segment** as the host video segment with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Documentary	37–101	0.95	25	23.16	31.13	0.8181	76.48
			50	23.16	32.35	0.8784	78.29
			75	23.16	32.5	0.889	78.59
			100	23.16	32.51	0.8922	78.67
		0.85	25	20	35.2	0.8924	72.77
			50	20	40.55	0.9631	75.88
			75	20	42.63	0.9771	76.75
			100	20	43.47	0.9817	77.07
		0.75	25	17.65	35.7	0.9046	68.31
			50	17.65	41.9	0.9727	71.56
			75	17.65	45.14	0.9863	72.71
			100	17.65	46.98	0.991	73.29

Table 8. Detailed results of the proposed Octree-hiding technique using the **Documentary medium-motion segment** as the host video segment with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Documentary	3181–3245	0.95	25	23.16	30.28	0.829	76.54
			50	23.16	31.34	0.8859	78.22
			75	23.16	31.47	0.8959	78.51
			100	23.16	31.48	0.8989	78.58
		0.85	25	20	34.61	0.9026	72.88
			50	20	39.32	0.9681	75.7
			75	20	40.93	0.9811	76.42
			100	20	41.54	0.9854	76.68
		0.75	25	17.65	35.29	0.9141	68.44
			50	17.65	41.05	0.9757	71.43
			75	17.65	43.74	0.988	72.41
			100	17.65	45.1	0.9922	72.85

In contrast, the "Monitoring" medium-motion segment (Table 5) shows a similar pattern where lower capacities yield better PSNR. When $C = 0.75$ and $k = 100$, the PSNR is 32.28 dB at a capacity of 17.65 bpp. This reinforces the notion that reducing the embedding rate enhances video quality, albeit the improvement is less pronounced than in slow-motion segments.

The "Monitoring" fast-motion segment (Table 6) further demonstrates this trade-off. Here, the highest PSNR achieved is 31.05 dB at a capacity of 17.65 bpp and $C = 0.75$, compared to 24.33 dB at 23.16 bpp. Fast-motion segments generally exhibit lower PSNR values due to having lower redundant data in the video, which makes data-hiding more challenging.

Table 9. Detailed results of the proposed Octree-hiding technique using the **Documentary fast-motion segment as the host video** segment with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Documentary	8053–8117	0.95	25	23.16	26.01	0.8136	75.08
			50	23.16	25.98	0.8201	75.24
			75	23.16	25.9	0.8184	75.17
			100	23.16	25.84	0.8167	75.12
		0.85	25	20	32.01	0.9424	73.23
			50	20	32.82	0.9603	73.88
			75	20	32.78	0.9622	73.91
			100	20	32.7	0.9623	73.9
		0.75	25	17.65	34.31	0.9601	69.35
			50	17.65	36.24	0.9797	70.32
			75	17.65	36.37	0.9824	70.42
			100	17.65	36.31	0.983	70.42

Table 10. Results of the proposed Octree technique when using **Film slow-motion segment as the host video segment** with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Film	1008–1072	0.95	25	23.16	30.87	0.8213	76.49
			50	23.16	32.4	0.888	78.54
			75	23.16	32.67	0.9005	78.92
			100	23.16	32.74	0.9043	79.03
		0.85	25	20	34.72	0.8936	72.68
			50	20	40.55	0.9699	76.05
			75	20	43.42	0.9855	77.15
			100	20	44.94	0.9907	77.67
		0.75	25	17.65	35.21	0.9038	68.17
			50	17.65	41.5	0.9744	71.51
			75	17.65	45.17	0.9888	72.78
			100	17.65	47.66	0.9938	73.53

In summary, across all video segments and motion types, there is a clear inverse relationship between embedding capacity (bpp) and video quality (PSNR). Lower embedding rates consistently yield higher PSNR values, underscoring the importance of optimizing the balance between the data-hiding capacity and resultant video quality. This trend holds true across various types of motion and video segments, validating the effectiveness of the proposed Octree-hiding technique.

For the Documentary slow-motion segment (Table 7), a significant improvement in PSNR is noted with decreasing bpp. The PSNR reaches 46.98 dB at the lowest capacity of 17.65 bpp and $C = 0.75$. This suggests that the proposed technique is highly effective in maintaining video quality at lower embedding rates for slow-motion content due to the high redundant information data available, which can be exploited for efficient data-hiding.

Table 11. Detailed results of the proposed Octree-hiding technique using **Film medium-motion segment as the host video segment** with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Film	1–64	0.95	25	23.16	33.16	0.8458	77.68
			50	23.16	34.43	0.9032	79.43
			75	23.16	34.61	0.9155	79.78
			100	23.16	34.64	0.9199	79.9
		0.85	25	20	36.65	0.8946	73.19
			50	20	41.53	0.9629	76.12
			75	20	43.59	0.9788	77.03
			100	20	44.46	0.9847	77.4
		0.75	25	17.65	36.95	0.9029	68.58
			50	17.65	42.79	0.9694	71.7
			75	17.65	45.84	0.9847	72.85
			100	17.65	47.55	0.9905	73.42

Table 12. Detailed results of the proposed Octree-hiding technique using **Film fast-motion segment as the host video segment** with a 32 × 32 × 32 minimum cube size.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Film	1632–1696	0.95	25	23.16	30.46	0.8307	76.62
			50	23.16	31.73	0.8939	78.52
			75	23.16	31.94	0.9059	78.87
			100	23.16	31.98	0.9094	78.97
		0.85	25	20	34.65	0.9009	72.85
			50	20	39.98	0.9708	75.93
			75	20	42.23	0.985	76.84
			100	20	43.26	0.9899	77.23
		0.75	25	17.65	35.24	0.9108	68.35
			50	17.65	41.4	0.9761	71.52
			75	17.65	44.8	0.9893	72.7
			100	17.65	46.95	0.9939	73.36

In the Documentary medium-motion segment (Table 8), the PSNR values follow the established pattern of higher PSNR at lower capacities. The PSNR at 17.65 bpp is 45.10 dB when $C = 0.75$, indicating good performance even in medium-motion contexts.

Finally, the Documentary fast-motion segment (Table 9) continues to show that lower capacities result in better video quality. The highest PSNR is 36.31 dB at 17.65 bpp, with $C = 0.75$, highlighting the consistent effectiveness of the technique across different motion types and capacities.

Table 10 presents the results for the Film slow-motion segment. As the scale k and capacity (bpp) increase, the PSNR values generally improve. For instance, with a scale of 100 and a capacity of 23.16 bpp, the PSNR is 32.74 dB. In contrast, a lower capacity of 17.65 bpp at the same scale results in a PSNR of 47.66 dB. This demonstrates a positive

correlation between bpp and PSNR; higher capacity enables better quality representation, although with diminishing returns at higher bpp values.

Table 11 details the Film medium-motion segment. The trend observed is similar to that in the slow-motion segment: higher bpp values correspond to higher PSNR values. Specifically, with 100 bpp, the PSNR reaches 34.64 dB, whereas at 17.65 bpp, the PSNR is 47.55 dB. This suggests that, regardless of motion speed, increasing the bit rate improves quality. The medium-motion segment shows slightly higher PSNR for the same bpp values compared to the slow-motion segment, indicating that medium-motion scenes might handle data embedding more efficiently.

Table 12 illustrates the performance of the film's fast-motion segment. The results confirm the observed trend, where PSNR increases with bpp. For example, at 100 bpp, the PSNR is 31.98 dB, compared to 46.95 dB at 17.65 bpp. The PSNR values are lower across the board in the fast-motion segment compared to the slow and medium-motion segments, which may be attributed to the challenges of maintaining visual quality in high-motion scenes.

The C threshold is the normalized version of the C' parameter shown in Equation (7). The parameter C in Tables 4–6, Tables 7–9, and Tables 10–12 is calculated as a percentage with respect to the total size of the video segment. Therefore, $C = 1$ refers to hiding at full capacity. Tables 4–6, Tables 7–9, and Tables 10–12 show clearly that increasing the C parameter increases the hiding capacity, but of course at an adverse effect on the quality of the carrier signal.

Tables 4–6, Tables 7–9, and Tables 10–12 show that by increasing the scaling factor k, the quality of the carrier video segment improves. The same justification is valid here also. Having a low scaling factor k will rescale the secret signal to larger scales than using high scaling factors. Since embedding can be treated as adding noise to the host signal, it is expected that adding a relatively larger-scaled signal will introduce some artifacts in the carrier signal. Therefore, in contrast with the T_{peak} parameter, the quality improves as the scaling factor k increases ($k \to \infty$). Figure 7 shows the effect of tuning the scaling factor k on the quality of the carrier video segment. However, as will be shown later in the robustness Section 4.1, this quality improvement has an adverse effect on the robustness of the proposed scheme.

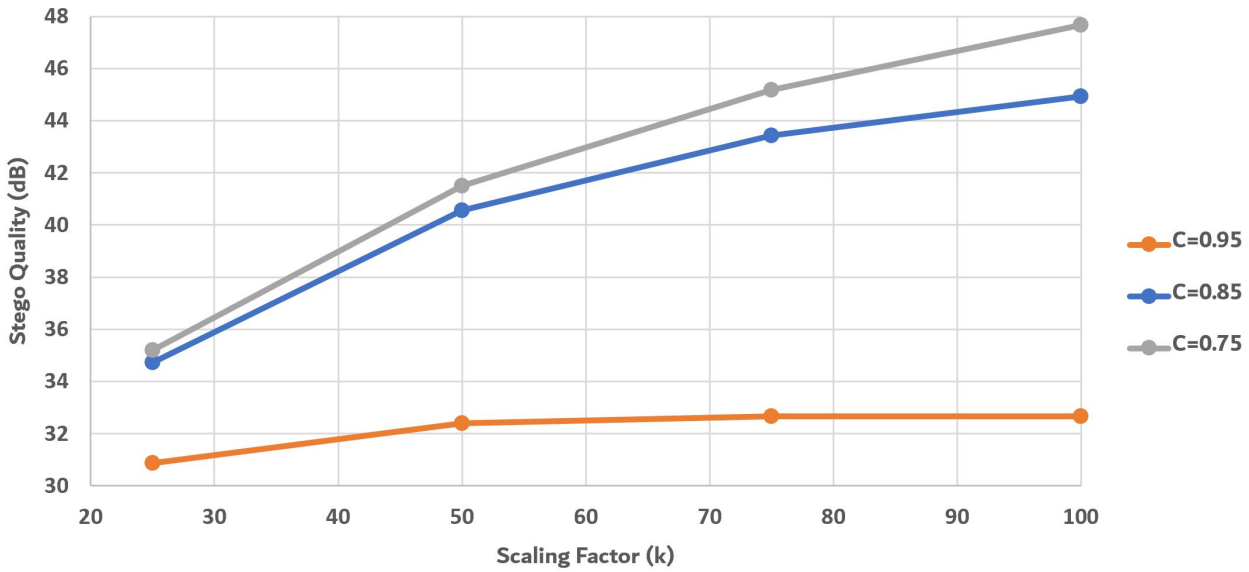


Figure 7. Illustrating the effect of the scaling factor k on the embedding capacity when hiding in the "F" slow motion segment.

In the Octree segmentation technique, there is another critical parameter that needs to be studied, which is the minimum cube size. Therefore, another experiment was performed to examine the effect of tuning the minimum cube size parameter. Tables 13–15 present the results obtained when hiding using different sizes of the allowed minimum cube sizes. Tables 13–15 show that increasing the cube sizes, increases the hiding capacity. The reason is that by reducing the size of the minimum cube size the number of cubes increases and, therefore, each cube will produce some significant DCT coefficients to maintain the structure of the cube to be able to regenerate the spatial representation with minor degradations. Naturally, increasing the number of significant DCT coefficients will reduce the number of insignificant DCT coefficients, and therefore, reduced capacity rates will be achieved.

Table 13. Detailed results of the proposed Octree-hiding technique using the **Documentary slow-motion segment** as the host video segment with different minimum cube sizes.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Documentary	1008–1072	8	25	12.75	37.28	0.9219	58.93
			50	12.75	43.42	0.9778	61.86
			75	12.75	46.73	0.9899	62.99
			100	12.75	48.75	0.9942	63.61
		16	25	21.61	33.29	0.8542	74.7
			50	21.61	35.79	0.9316	77.26
			75	21.61	36.28	0.9482	77.79
			100	21.61	36.4	0.9537	77.96
		32	25	23.63	28.36	0.7479	75.01
			50	23.63	28.88	0.7937	76.28
			75	23.63	28.91	0.8009	76.47
			100	23.63	28.9	0.8026	76.51

Table 14. Detailed results of the proposed Octree-hiding technique using the **Documentary medium-motion segment** as the host video segment with different minimum cube sizes.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Documentary	3181–3245	8	25	12.75	37.15	0.9258	59
			50	12.75	43.19	0.9794	61.85
			75	12.75	46.31	0.9907	62.91
			100	12.75	48.07	0.9946	63.44
		16	25	21.61	32.87	0.8689	74.96
			50	21.61	35.08	0.9387	77.26
			75	21.61	35.47	0.9535	77.72
			100	21.61	35.56	0.9583	77.87
		32	25	23.63	27.3	0.7421	74.6
			50	23.63	27.74	0.784	75.76
			75	23.63	27.77	0.7901	75.92
			100	23.63	27.77	0.7912	75.94

Table 15. Detailed results of the proposed Octree-hiding technique using the **Documentary fast-motion segment** as the host video segment with different minimum cube sizes.

Host	Frames	*C*	Scale *k*	Capacity (bpp)	PSNR (dB)	SSIM	CCQE (%)
Documentary	37–101	8	25	12.75	36.51	0.9719	59.99
			50	12.75	39.95	0.9903	61.31
			75	12.75	40.65	0.9933	61.56
			100	12.75	40.78	0.9941	61.61
		16	25	21.61	29.09	0.9006	74.81
			50	21.61	29.38	0.918	75.31
			75	21.61	29.31	0.9191	75.32
			100	21.61	29.24	0.9186	75.3
		32	25	23.63	23.92	0.717	73.13
			50	23.63	23.85	0.7177	73.13
			75	23.63	23.77	0.7143	73.02
			100	23.63	23.73	0.7118	72.95

The advantage on the other hand of using a small minimum cube size is the enhanced quality of the carrier signal generated by the proposed Octree-hiding scheme. Therefore, the minimum cube size must be elected carefully based on the intended goals whether increased capacity rates or improved carrier quality.

4.1. Robustness Against Tampering Attack

This section evaluates the robustness of the proposed technique against tampering attacks. As shown in Figure 8, the extracted secret image demonstrates the effects of tampering on the carrier video segment. The relationship between capacity, robustness, and quality can be visualized as a trade-off triangle; enhancing one aspect negatively impacts the other two. Typically, watermarking schemes prioritize the robustness-quality trade-off, whereas steganography schemes focus on the capacity–quality trade-off. Although the proposed technique primarily aims to embed large secret messages while maintaining a satisfactory carrier quality, the robustness tests indicate that it performs acceptably in comparison to the significantly high capacity ratio achieved by the proposed Octree technique.

The first test to be performed in this section is to explore the quality of the extracted image without performing attacks on the carrier signal. Figure 9 shows the PSNR values between the recovered image and the original secret image (orange curve), and the PSNR values between host and carrier video segments (blue curve) when testing using a range of scaling factors k = [1:100].

Figure 9 clearly shows that the extraction quality is affected by the scaling factor *k*. Based on Figure 9, reducing the scaling factor *k* enhances the carrier quality. The reason is that the embedding process can be treated as a noise insertion process. Based on Equation (9), increasing the scaling factor *k* will reduce the amplitude of the secret signal. Therefore, enhanced carrier quality can be achieved, as shown in Figure 9.

On the other hand, Figure 9 shows that reducing the value of the scaling factor *k* enhances the extraction quality. The justification can be extracted also from Equation (9). By increasing the scaling factor *k*, the amplitude of the secret signal will be minimized. The carrier video segments are encoded using an 8-bit integer system. However, the DCT domain is usually represented in decimals. Therefore, the inverse DCT (IDCT) will produce a floating-point matrix which later on should be rounded to the nearest integer. If the secret

data are rescaled to a low scale, then rounding would destroy the secret image. Figure 9 shows that the range $k = [10 - 100]$ is safe for hiding since results are above 20 dB which is an acceptable quality level between images [35].

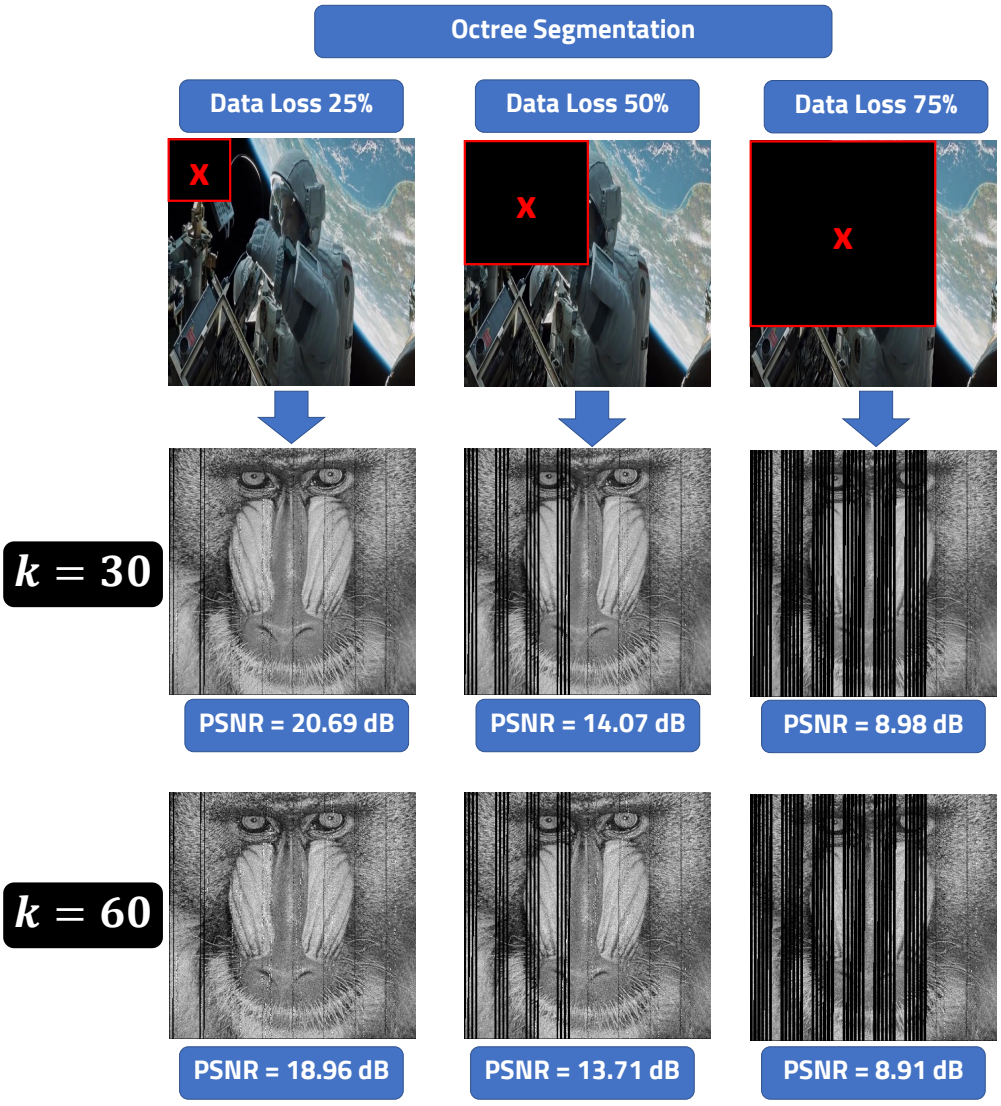


Figure 8. Examining the robustness of the proposed Octree technique against data tampering. The percentages shown in the figure are calculated based on the dimension of the tampered cube with respect to the dimension of the carrier frame.

Figure 8 shows the extraction results after attacking the carrier image with geometric attacks (cropping or data tampering). A data tampering attack is done by removing some part (a cube for example) of the carrier signal (by replacing it with zeros). The percentage that appears in Figure 8 refers to the dimension of the cropping cube with respect to the dimension of the carrier signal. For example, if the carrier video segment has a dimension of $512 \times 512 \times 512$, then a data loss of 50% means cropping the carrier video segment with a $256 \times 256 \times 256$ cube.

Based on Figure 8, the proposed Octree-hiding technique has achieved acceptable recovering results. The reason is that the proposed Octree segmentation technique segments the host into cubes, and based on Figure 8 tampering is done in a cube in the top-left corner of the carrier signal. Therefore, any hidden data within that specific block will be destroyed. On the other hand, data that are embedded in other cubes will be intact.

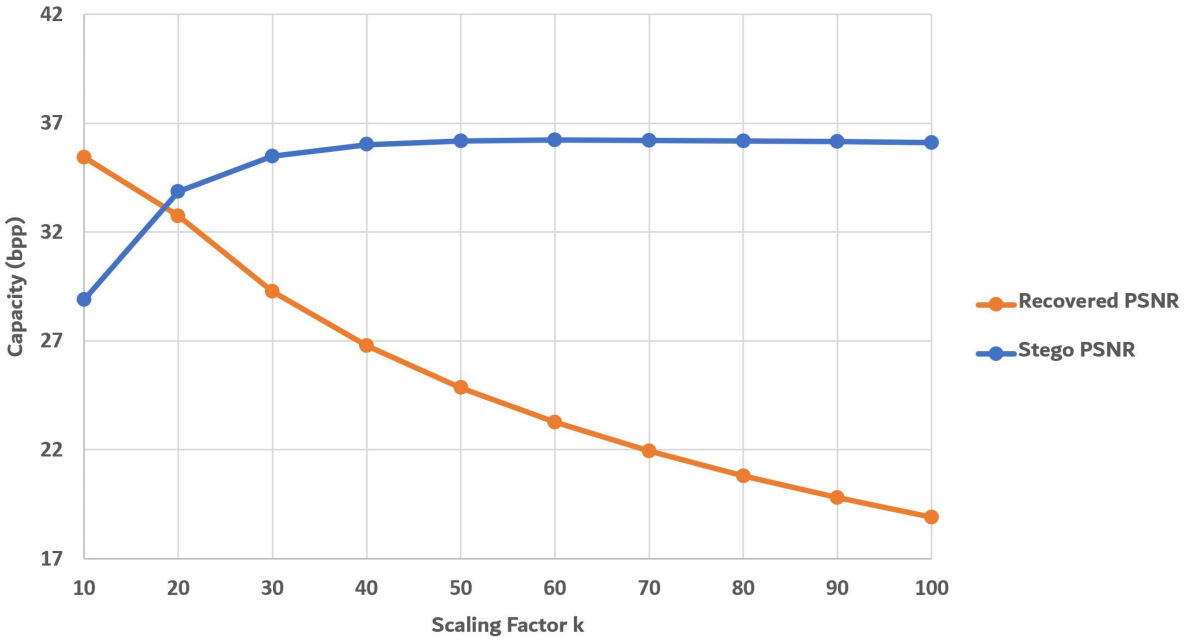


Figure 9. Illustrating the effect of tuning the scaling factor *k* on the quality of the extracted image and the quality of the carrier video segment. Tests were performed using the proposed Octree-hiding technique when hiding using the Documentary host video segment.

4.2. Steganalysis Test

This section discusses the security level of the proposed hiding schemes. In the context of steganography, a hiding scheme is deemed compromised if a third party can reveal the existence of the secret data. The process of detecting hidden data within a carrier is known as steganalysis. The primary goal of a steganalysis tool is to distinguish between the host and carrier signals. Steganalysis techniques are generally classified into three categories: statistical-based methods, signature-based methods, and machine learning methods.

Statistical-based methods analyze the statistical properties of the carrier signal to identify hidden data. In contrast, signature-based methods focus on identifying repetitive patterns within the carrier signal that are a result of the steganography technique. Machine learning methods, on the other hand, learn to classify host and carrier signals by being trained on a set of handcrafted features. Recently, with the rise of deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), these methods have also been employed for steganalysis. The main advantage of deep learning is its ability to automatically extract features from the training data, eliminating the need for manual feature engineering.

The RNNs are a class of neural networks designed to process sequential data, making them particularly effective for capturing time-based signals. Unlike traditional feedforward networks, RNNs maintain a hidden state that retains information from previous time steps, allowing them to learn temporal dependencies and patterns within the data. This capability enables RNNs to analyze sequences, such as time series data, natural language, or audio signals, by effectively remembering past inputs while incorporating new information as it becomes available. Therefore, in this work, a long short-term memory (LSTM) network, which is an RNN-based network, is trained to detect and classify temporal signals to be either a host signal or carrier signal. Since a video segment is a spatiotemporal signal, the input data fed into the LSTM network are the features of each frame obtained by a well-known CNN network named GoogLeNet https://www.mathworks.com/help/deeplearning/ref/googlenet.html (accessed on 19 December 2024).

The neural network architecture is specifically engineered to process sequence data, comprising several distinct layers each serving a crucial function. The first layer is a sequence input layer designed to handle input with 'n' dimensions, where 'n' represents the number of samples in the input signal. This allows the network to process varying lengths of sequence data effectively. Following the input layer, the network features a bidirectional long short-term memory (BiLSTM) layer, which is configured with 2000 hidden units. The BiLSTM layer is pivotal for the model as it processes data from both forward and backward directions, capturing dependencies and patterns that span across time in the sequence data. This dual-direction processing significantly enhances the network's ability to understand and predict based on the temporal context inherent in the input sequences.

Subsequent to the BiLSTM layer, the architecture includes two fully connected layers that serve to synthesize the features extracted by the BiLSTM, facilitating the transition from feature extraction to decision-making processes within the network. These layers are crucial for integrating learned temporal features and preparing the network output for classification. The network employs a softmax layer as the penultimate step, which maps the output of the fully connected layers onto a probability distribution over predicted output classes. This is essential for tasks such as classification, where the output needs to represent a probability that the input sequence belongs to a particular class. Finally, the architecture concludes with a classification output layer, which formalizes the network's predictions based on the probabilities calculated by the softmax layer. This layer effectively determines the most likely class for the given input based on the learned temporal dependencies and patterns.

The proposed technique has achieved high hiding capacity ratios, making it more susceptible to detection by steganalysis tools. However, as shown in Figure 10, The RNN was unable to detect the secret data within the carrier signals produced by the proposed Octree technique. The accuracy rate of the proposed hiding strategy stands at 50%, suggesting that the RNN could not effectively differentiate between the host and carrier signals.

An additional steganalysis test was performed using the OpenPuff v4.01 tool to evaluate the robustness of the proposed Octree technique. Figure 11 shows the analysis results for all stego video segments generated by the technique. The tool failed to detect any hidden secret data in all the files analyzed. For every file, the "Mean Integrity" was reported as 0.00%, and the "Mark" status was consistently identified as "not a string", indicating the absence of detectable embedded markers or hidden content. Additionally, the "Overall Integrity" of 100.00% across all files confirms that the proposed Octree technique preserves the visual and structural integrity of the stego video segments. These findings demonstrate the effectiveness of the proposed method in securely concealing secret data, ensuring it remains undetectable to one of the most widely used steganalysis tools, further validating the robustness and stealthiness of the technique.

The enhanced security of the proposed Octree technique is largely attributed to the spatiotemporal hiding strategy employed within the method. Unlike previous methods, such as those outlined in [8], which utilize 1D vectors for hiding, the proposed Octree technique implements a 3D segmentation technique that operates across three-dimensional regions. This approach allows for the distribution of noise along the edges of these regions, which are high-frequency areas of the image. The literature indicates that the human visual system is less sensitive to disturbances in these high-frequency zones, thereby reducing the visibility of any abrupt visual discrepancies between regions [36].

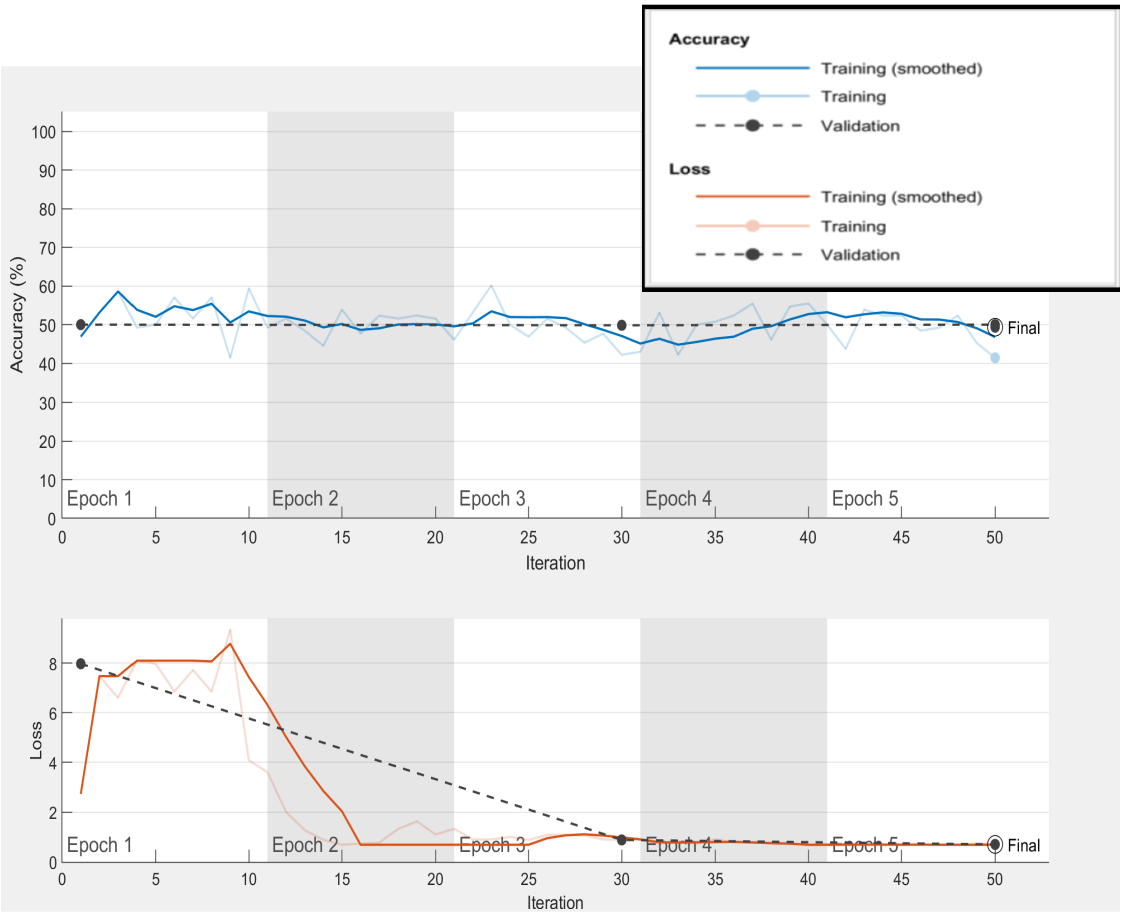


Figure 10. Steganalysis test: Attempting to distinguish between a host signal and an Octree carrier signal. As it is clear, the RNN failed to classify the images.

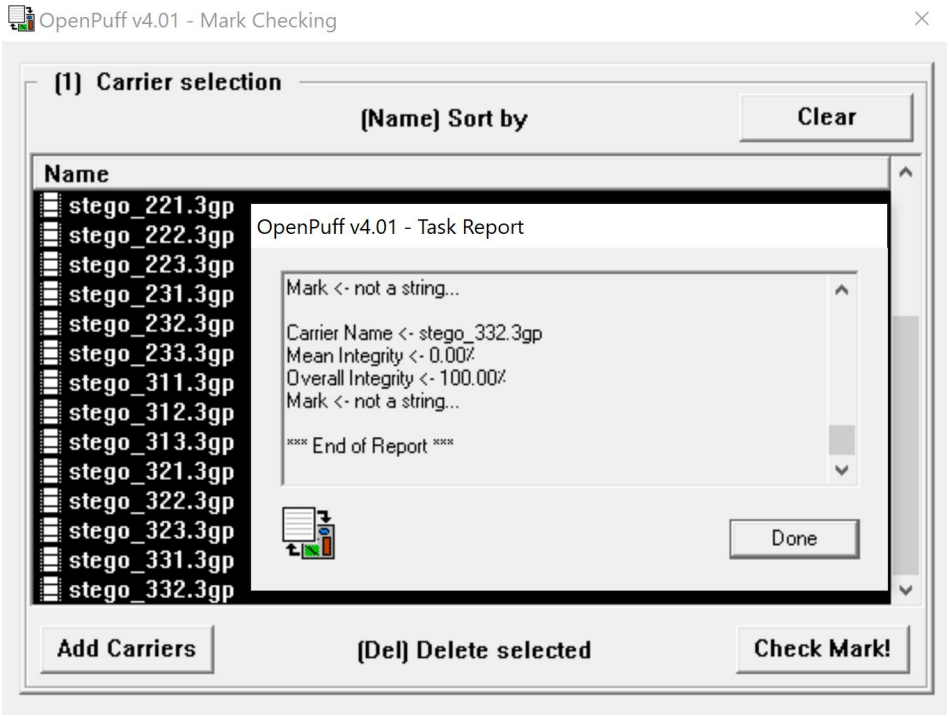


Figure 11. Results of the OpenPuff steganalysis test on stego video segments produced by the proposed Octree technique, showing no detectable hidden data and 100% overall integrity across all files.

4.3. Computational Complexity

Regarding the computational complexity, the proposed embedding scheme includes two key loops. The aim of the first for-loop is to determine the total number of insignificant coefficients to determine the hiding capacity ratio. The second loop embeds the secret image in the DCT cubes of the cover signal.

If it is presumed that T_1 is the execution time for every iteration in the first for-loop, T_2 is the execution time for every iteration in the second for-loop, and N is the total number of cubes, the computational complexity of the embedding process is:

$$O([T_1 + T_2] \times N) = O(N) \tag{10}$$

The execution time and the total number of regions created for every cover image are proportional as can be seen in Equation (10). Therefore, the execution time increases linearly with the total number of segmented regions.

However, compared to the technique proposed in [32], which had the same Big-O complexity as shown in Equation (10), the proposed Octree technique demonstrated faster execution speed. For a $256 \times 256 \times 256$ video segment processed using an Intel Xeon 2.9 GHz processor with 32 GB of RAM, the Octree technique achieved an average speed of 16.7 s, while the directional scheme required 272.9 s. This improvement is due to Octree's use of cubes, as opposed to the pixel-by-pixel processing approach of the directional scheme.

The execution speed of the proposed Octree technique can be further improved through parallel processing. By dividing the video segment into smaller regions and processing these regions simultaneously, the computational workload can be distributed across multiple cores or processors. This approach is particularly well-suited for the Octree technique, as its hierarchical structure and cube-based processing naturally lend themselves to parallelization.

For instance, each level of the Octree or each sub-cube can be processed independently, significantly reducing the overall execution time compared to sequential processing. With modern hardware, such as multi-core CPUs or GPUs, parallel implementation could drastically enhance the performance, potentially bringing the execution time well below the 16.7 s achieved on a single-threaded Intel Xeon 2.9 GHz processor. This scalability highlights the potential of the Octree technique to handle larger video segments or real-time applications without sacrificing efficiency.

5. Future Directions

To address the evolving challenges and opportunities in steganography, future research must consider the impact of emerging technologies such as quantum computing and brain-like intelligence, particularly in relation to the proposed Octree technique. The hierarchical structure and cube-based segmentation of the Octree technique could be well-suited for adaptation to quantum computing, which thrives on efficient data organization and processing. Quantum algorithms could enhance the Octree technique by optimizing the embedding and extraction processes, significantly reducing execution time while increasing the capacity and robustness of the hidden data. However, adapting the Octree approach to quantum systems would require rethinking certain elements, such as incorporating quantum error correction and quantum-resistant cryptographic schemes. Future studies could investigate how Octree's segmentation and data compaction principles align with the computational models of quantum systems, identifying which classical processes can be retained and which need modification to fully harness quantum computing's potential [37].

Brain-like intelligence and advanced AI technologies also present transformative opportunities for the Octree technique. The hierarchical and spatially adaptive nature of

the Octree framework makes it compatible with AI-driven analysis, which can process data with high complexity and correlation. AI models inspired by brain-like intelligence could further enhance the detection and embedding processes by leveraging their ability to analyze multi-dimensional data. Conversely, the Octree technique itself could inspire new steganographic methods aimed at evading AI detection. For example, the technique's ability to exploit insignificant coefficients in highly correlated regions could be adapted to embed data that remains imperceptible to even the most advanced AI models. This could lead to the development of "AI-targeted steganography", where the Octree approach is refined to interact specifically with the ways AI systems perceive and process data [38,39].

By extending the Octree technique into these emerging domains, steganography can evolve to meet the challenges of a rapidly changing technological landscape. The hierarchical structure and adaptability of the Octree provide a strong foundation for integrating quantum and AI-based advancements, ensuring the technique remains relevant and effective in an increasingly AI-driven and quantum-enabled world. These future directions highlight the potential of the Octree technique not only as a robust steganographic tool but also as a platform for exploring new intersections between steganography and cutting-edge technologies.

6. Conclusions

This paper introduces a novel approach for leveraging spatiotemporal redundancy to conceal secret data within video segments. Videos inherently contain significant data redundancy from both spatial and temporal dimensions, making it advantageous to exploit both for optimal results. This is achieved by dividing the host video into non-overlapping homogeneous cubes based on inter-pixel intensity values. Data-hiding occurs in the less significant DCT coefficients of each cube. Due to the internal correlations within the cubes, the energy concentration is enhanced, resulting in a high number of insignificant DCT coefficients available for embedding and, consequently, higher embedding ratios. The experimental results demonstrate that the proposed Octree steganography technique surpasses several recent video-hiding methods in terms of embedding ratio and carrier quality. Future research will investigate the Octree technique's robustness and the quality of extracted data. Additionally, a thorough analysis will be conducted to comprehensively assess the performance in terms of capacity and quality. Moreover, audio steganography will be integrated with the proposed Octree technique to develop a comprehensive 4D audio-video steganography solution.

Author Contributions: Conceptualization, M.B., T.R. and I.K.; methodology, M.B., T.R. and I.K.; software, M.B.; validation, M.B.; writing—review and editing, M.B.; supervision, M.B., T.R. and I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Baziyad, M.; Rabie, T.; Kamel, I.; Benkhelifa, M. Polynomial fitting: Enhancing the stego quality of DCT-based Steganography schemes. *Multimed. Tools Appl.* **2022**, *81*, 43999–44019. [CrossRef]
2. Yao, Y.; Wang, J.; Chang, Q.; Ren, Y.; Meng, W. High invisibility image steganography with wavelet transform and generative adversarial network. *Expert Syst. Appl.* **2024**, *249*, 123540. [CrossRef]
3. Kombrink, M.H.; Geradts, Z.J.M.H.; Worring, M. Image steganography approaches and their detection strategies: A survey. *Acm Comput. Surv.* **2024**, *57*, 1–40. [CrossRef]

4. Su, W.; Ni, J.; Hu, X.; Li, B. Efficient Audio Steganography Using Generalized Audio Intrinsic Energy With Micro-Amplitude Modification Suppression. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 6559–6572. [CrossRef]
5. Dasgupta, K.; Mandal, J.; Dutta, P. Hash based least significant bit technique for video steganography (HLSB). *Int. J. Secur. Priv. Trust. Manag. (IJSPTM)* **2012**, *1*, 1–11.
6. Rabie, T.; Baziyad, M.; Kamel, I. High-Fidelity Steganography: A Covert Parity Bit Model-Based Approach. *Algorithms* **2024**, *17*, 328. [CrossRef]
7. Baziyad, M.; Rabie, T.; Kamel, I. Toward stronger energy compaction for high capacity dct-based steganography: A region-growing approach. *Multimed. Tools Appl.* **2020**, *80*, 8611–8637 [CrossRef]
8. Rabie, T.; Baziyad, M. The pixogram: Addressing high payload demands for video steganography. *IEEE Access* **2019**, *7*, 21948–21962. [CrossRef]
9. Sadek, M.M.; Khalifa, A.S.; Mostafa, M.G. Video steganography: A comprehensive review. *Multimed. Tools Appl.* **2015**, *74*, 7063–7094. [CrossRef]
10. Shirali-Shahreza, M. A new method for real-time steganography. In Proceedings of the 2006 8th international Conference on Signal Processing, Guilin, China, 16–20 November 2006; Volume 4.
11. Gilkarov, D.; Dubin, R. Steganalysis of AI Models LSB Attacks. *IEEE Trans. Inf. Forensics Secur.* **2024**, 19, 4767–4779. [CrossRef]
12. Dasgupta, K.; Mondal, J.K.; Dutta, P. Optimized video steganography using genetic algorithm (GA). *Procedia Technol.* **2013**, *10*, 131–137. [CrossRef]
13. Singh, S.; Agarwal, G. Hiding image to video: A new approach of LSB replacement. *Int. J. Eng. Sci. Technol.* **2010**, *2*, 6999–7003.
14. Chang, K.C.; Chang, C.P.; Huang, P.S.; Tu, T.M. A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing. *J. Multimed.* **2008**, *3, 37–44*. [CrossRef]
15. Sherly, A.; Amritha, P. A compressed video steganography using TPVD. *Int. J. Database Manag. Syst. (IJDMS)* **2010**, *2*, 67–80.
16. Mstafa, R.J.; Elleithy, K.M. A high payload video steganography algorithm in DWT domain based on BCH codes (15, 11). In Proceedings of the Wireless Telecommunications Symposium (WTS), 2015, New York City, NY, USA, 15–17 April 2015; pp. 1–8.
17. Sadek, M.M.; Khalifa, A.S.; Mostafa, M.G. Robust video steganography algorithm using adaptive skin-tone detection. *Multimed. Tools Appl.* **2017**, *76*, 3065–3085. [CrossRef]
18. Ke, N.; Weidong, Z. A video steganography scheme based on H. 264 bitstreams replaced. In Proceedings of the 2013 IEEE 4th International Conference on Software Engineering and Service Science, Beijing, China, 23–25 May 2013; pp. 447–450.
19. Liao, K.; Lian, S.; Guo, Z.; Wang, J. Efficient information hiding in H. 264/AVC video coding. *Telecommun. Syst.* **2012**, *49*, 261–269. [CrossRef]
20. Mansouri, J.; Khademi, M. An adaptive scheme for compressed video steganography using temporal and spatial features of the video signal. *Int. J. Imaging Syst. Technol.* **2009**, *19*, 306–315. [CrossRef]
21. Mstafa, R.J.; Elleithy, K.M.; Abdelfattah, E. A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC. *IEEE Access* **2017**, *5*, 5354–5365. [CrossRef]
22. Agrawal, V.K. Perceptual Watermarking of Digital Video Using the Variable Temporal Length 3d-Dct. Master's Thesis, Indian Institute of Technology, Kanpur, India, 2007.
23. Baziyad, M.; Shahin, I.; Rabie, T.; Nassif, A.B. Maximizing embedding capacity for speech steganography: A segment-growing approach. *Multimed. Tools Appl.* **2021**, *80*, 24469–24490. [CrossRef]
24. Baziyad, M.; Shahin, I.; Rabie, T.; Nassif, A.B. 64-bit quantization: Taking payload capacity of speech steganography to the limits. *Multimed. Tools Appl.* **2022**, *81*, 40561–40579. [CrossRef]
25. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete cosine transform. *IEEE Trans. Comput.* **1974**, *100*, 90–93. [CrossRef]
26. Baziyad, M.; Obaidat, M.S. On the Importance of the DCT Phase for Image Steganography Schemes. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 30–31 October 2020; pp. 791–795.
27. Lee, M.; Chan, R.K.; Adjeroh, D.A. Quantization of 3D-DCT coefficients and scan order for video compression. *J. Vis. Commun. Image Represent.* **1997**, *8*, 405–422. [CrossRef]
28. Rabie, T.; Baziyad, M.; Kamel, I. Enhanced high capacity image steganography using discrete wavelet transform and the Laplacian pyramid. *Multimed. Tools Appl.* **2018**, 77, 23673–23698. [CrossRef]
29. Rabie, T.; Baziyad, M.; Bonny, T.; Fareh, R. Toward a Unified Performance Metric for Benchmarking Steganography Systems. *J. Circuits Syst. Comput.* **2020**, *29*, 2050042. [CrossRef]
30. Xu, D.; Wang, R.; Shi, Y.Q. Data hiding in encrypted H. 264/AVC video streams by codeword substitution. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 596–606. [CrossRef]
31. Mstafa, R.J.; Elleithy, K.M. A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes. *Multimed. Tools Appl.* **2016**, *75*, 10311–10333. [CrossRef]

32. Baziyad, M.; Rabie, T.; Kamel, I. Directional Pixogram: A New Approach for Video Steganography. In Proceedings of the 2020 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 4 February–9 April 2020; pp. 1–5.
33. Vakani, H.; Abdallah, S.; Kamel, I.; Rabie, T.; Baziyad, M. Dct-in-dct: A novel steganography scheme for enhanced payload extraction quality. In Proceedings of the 2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bandung, Indonesia, 27–28 July 2021; pp. 201–206.
34. Rabie, T.; Baziyad, M.; Kamel, I. Secure high payload steganography: A model-based approach. *J. Inf. Secur. Appl.* **2021**, *63*, 103043. [CrossRef]
35. Rabie, T.; Kamel, I. Toward optimal embedding capacity for transform domain steganography: A quad-tree adaptive-region approach. *Multimed. Tools Appl.* **2017**, *76*, 8627–8650. [CrossRef]
36. Baziyad, M.; Rabie, T.; Kamel, I. Extending steganography payload capacity using the L* a* b* color space. In Proceedings of the 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 18–19 November 2018; pp. 1–6.
37. He, J.; Zhu, H.; Zhou, X. Quantum image encryption algorithm via optimized quantum circuit and parity bit-plane permutation. *J. Inf. Secur. Appl.* **2024**, *81*, 103698. [CrossRef]
38. Chaumont, M. Deep learning in steganography and steganalysis. In *Digital Media Steganography*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 321–349.
39. Yi, Z.; Lian, J.; Liu, Q.; Zhu, H.; Liang, D.; Liu, J. Learning rules in spiking neural networks: A survey. *Neurocomputing* **2023**, *531*, 163–179. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.