

Article

Computer Network Redundancy Reduction Using Video Compression

Shabana Habib ^{1,*}, Waleed Albattah ¹, Mohammed F. Alsharekh ², Muhammad Islam ³,
Mohammad Munawar Shees ³ and Hammad I. Sherazi ²

¹ Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia; w.albattah@qu.edu.sa

² Department of Electrical Engineering, Unaizah College of Engineering, Qassim University, Buraydah 51452, Saudi Arabia; m.alsharekh@qu.edu.sa (M.F.A.); h.sherazi@qu.edu.sa (H.I.S.)

³ Department of Electrical Engineering, College of Engineering and Information Technology, Onaizah Colleges, Onaizah 56447, Saudi Arabia; m.islam@oc.edu.sa (M.I.); mmshees@gmail.com (M.M.S.)

* Correspondence: s.habibullah@qu.edu.sa

Abstract: Due to the strong correlation between symmetric frames, video signals have a high degree of temporal redundancy. Motion estimation techniques are computationally expensive and time-consuming processes used in symmetric video compression to reduce temporal redundancy. The block-matching technique is, on the other hand, the most popular and efficient of the different motion estimation and compensation techniques. Motion compensation based on the block-matching technique generally uses the minimization of either the mean square error (MSE) or mean absolute difference (MAD) in order to find the appropriate motion vector. This paper proposes to remove the highly temporally redundant information contained in each block of the video signal using the removing temporal redundancy (RTR) technique in order to improve the data rate and efficiency of the video signal. A comparison between the PSNR values of this technique and those of the JPEG video compression standard is made. As a result of its moderate memory and computation requirements, the algorithm was found to be suitable for mobile networks and embedded devices. Based on a detailed set of testing scenarios and the obtained results, it is evident that the RTR compression technique allowed a compression ratio of 22.71 and 95% loss in bit rate reduction while maintaining sufficient intact signal quality with minimized information loss.

Keywords: temporal redundancy; symmetric motion estimation; block matching; mean absolute difference; peak signal noise ratio



Citation: Habib, S.; Albattah, W.; Alsharekh, M.F.; Islam, M.; Shees, M.M.; Sherazi, H.I. Computer Network Redundancy Reduction Using Video Compression. *Symmetry* **2023**, *15*, 1280. <https://doi.org/10.3390/sym15061280>

Academic Editors: Aviv Gibali and Alexander Zaslavski

Received: 9 February 2023

Revised: 5 April 2023

Accepted: 29 May 2023

Published: 19 June 2023



Copyright: © 2023 by the authors. 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

The immense growth of computer performance and declining costs of storage devices over the past decades have led to the dominance of multimedia data in cyberspace, increasing the volumes of transmitted data and repositories to support a wide range of symmetric and future applications in mobile communication networks. In order for videos to be transmitted and stored efficiently with less memory, video compression is essential [1,2]. As an example, a computer with a screen resolution of 1024 by 728 and a screen refresh rate of 75 Hz is capable of producing more than 100 MB of data every second [3], necessitating a large amount of storage against the normal practice of limited memory on mobile devices using multimedia platforms. Therefore, video compression has gained more attention due to the need for internet video streaming as well as for many other multimedia signals to be transmitted over a limited amount of bandwidth with a fixed number of video channels. In a forensic investigation, low-quality CCTV images are frequently enhanced for the extraction of potential evidence in digital forensic investigation [4].

Several symmetric and asymmetric studies have been performed on video transmission to improve theoretical and practical aspects of the field. The study of developing new

techniques for video compression is not only required for transmitting fast video signals over the internet but also for transmitting many other multimedia signals over a fixed number of limited-bandwidth channels. Fortunately, video contains a large amount of identically repetitive elements that can be excluded without losing the information content needed for human recognition capabilities [5].

Thus, through video compression, symmetrical data and lossless information have been discarded without affecting visual quality and acuity, using various algorithm-driven AI techniques in the literature [6].

Lossless data compression has been used to compress files to smaller sizes. As a result, software is often packaged before it is transmitted over the network to reduce the amount of bandwidth and time needed compared to it was transmitted as normal data files [7,8]. Lossless data compression has a limitation in that uncompressed data must be identical to the original data that was compressed in the first place. As a result of the compression scheme for lossy compression in real-time video compression and decompression according to the lossy compression scheme, some metadata are ignored when it comes to video compressions such as JPG and MPEG. Among the most widely used video compression techniques, motion estimation-based video encoding is commonly used. The idea is to reduce processing and computing overhead by performing compression during sensing to minimize the time needed for the compression of gathered data. Motion estimation-based video encoding is commonly used among the most widely used video compression techniques. The encoding method is based on asymmetrical frame correlation. The idea is to provide good compression and remove temporal redundancy through this method [9,10].

In another aspect, the motion estimation process is computationally expensive and very difficult to achieve in real time [11]. However, in order to achieve low-cost and efficiency in real time, the original video must be compressed before it is transmitted [12]. The videos are digitally compressed by encoding algorithms, representing them as a reduced set of bits before being transmitted over a network. Video compression uses compression standards such as MPEG-1, MPEG-2, MPEG-4, and H.264/AVC. The quality of the transmitted sequences is highly dependent on the bandwidth of the network; that is, the larger the bandwidth, the less effect on quality. The authors in [13] reported about the new H.264/AVC coding standard, which is a joint standard of the ITU-T video coding expert group (VCEG) and the ISO/IEC moving picture experts group (MPEG), incorporates many techniques to improve compression efficiency [14]. Similarly, compression algorithms are important in the case of limited resources and low data transmission capacity, which are affected by bandwidth and poor speed [15,16].

However, it is worth noting that even though modern cellular and wireless LAN channels provide sufficient bandwidth, no devices support video transmission. This is mainly due to the heavy computational burden imposed on video compression. Standard video compression systems such as JPG, MPEG-1, MPEG-2, and MPEG-4 all rely on complex motion estimation algorithms for video compression [17]. The computing power required to implement these algorithms is illustrated by the fact that only recently have personal computers been able to compress video in real time. A typical mobile device has much fewer computational resources than an ordinary computer [18]. Although custom video compression chipsets are available, they have not yet been included in general technology due to their cost, power, and size limitations [19,20]. A technical barrier has prevented the deployment of video-enabled devices in wireless and other mobile networks [21,22]. Video streaming over networks can cause bandwidth issues if sent in their redundancy-loudened form, as video streams are made from the temporal distribution of symmetrical frames similar to the special distribution of pixels in pictures. The pictures suffer from special redundancy due to asymmetrical frame-encoding techniques, similar to videos suffering from temporal redundancy due to what is called asymmetrical frame-encoding techniques [23,24]. Video temporal redundancy mitigation techniques have been widely developed since the early 1980s, such as the new three-step search (NTSS) algorithm for

motion-compensated asymmetrical frame coding for video conferencing [25], where the utility of the technique was shown by comparing it with contemporary techniques using a statistical parameter of mean square error (MSE). The block-based gradient descent search (BBGDS) approach was proposed in [26] and compared again in terms of MSE computational complexity using the concept of the motion vector. The distribution of motion vectors on several commonly experimented test image sequences was studied in [27], using a diamond search algorithm and a novel cross-diamond search algorithm for fast block-based motion estimation [28,29]. These studies tested simulations on hypothetical scenarios, such as that of the travel salesman problem.

The contributions of this paper can be summarized as follows:

- We proposed a scheme that avoids the use of redundant features, representing bottlenecks in auto-encoders.
- In this paper, we proposed explicitly penalizing pairwise correlations between features as part of the auto-encoder loss. We further developed a method to learn diverse compressive embeddings of samples based on the correlations.
- The proposed method was extensively evaluated on three tasks: dimensionality reduction, image compression, and image denoising. Compared to the standard approach, the results showed that the performance boost was consistent and measurable.

As outlined in this paper, one of the goals of this work was to overcome these limitations by abandoning the existing motion estimation and compensation model in order to provide a video compression algorithm that can be implemented by a mobile communication network in a real-time scenario. Currently, video compression techniques have not been able to adequately handle the high degree of temporal redundancy among symmetrical video frames. Indeed, the RTR video compression technique takes temporal redundancy impartially into account. To improve this technique's efficiency and data rate, temporally redundant video signals were discarded so that a minimal amount of complexity was introduced.

The rest of the sections of the paper are arranged as follows. The proposed approach is presented in Section 2 of this document. Section 3 of this paper presents the results and discusses the dimensionality reduction task using the removing temporal redundancy technique to compress images. In Section 4 of the paper, we conclude our discussion.

2. Proposed Approach

Compression is required in online video streaming and broadcast applications to transmit multimedia signals over a wireless network due to the fixed and limited amount of bandwidth available in a wireless network. Wireless communication systems face a big challenge in transmitting large video files. Therefore, video compression techniques save storage space by reducing the file size. Various techniques are used to find the redundancies in moving frames and correlations between the scenes. The main goal of the proposed technique was to compress the video using more efficient compression redundancy in the temporal domain in order to achieve improved compression results while maintaining the same acuity vision. This paper describes the application of RTR to check the performance of the proposed technique based on the compression ratio, as shown in Figure 1.

2.1. Encoder Buffer

The video is captured in real time and sent to the encoding buffer for conversion to several frames as the input of the RTR, which are then rendered as still images in JPEG format. All images have a number assigned to them. In the encoder, the 1st or previous frame is considered to be the current symmetric frame, while the 2nd or next frame is considered to be the reference frame. However, the encoder contains a sequence of frames as follows:

$$I(n, r) = I(n - 2), I(n - 1) \times I(n - r + 1) \text{ Next : encode } I(n) \quad (1)$$

where r represents distinct objects from n objects without replacement.

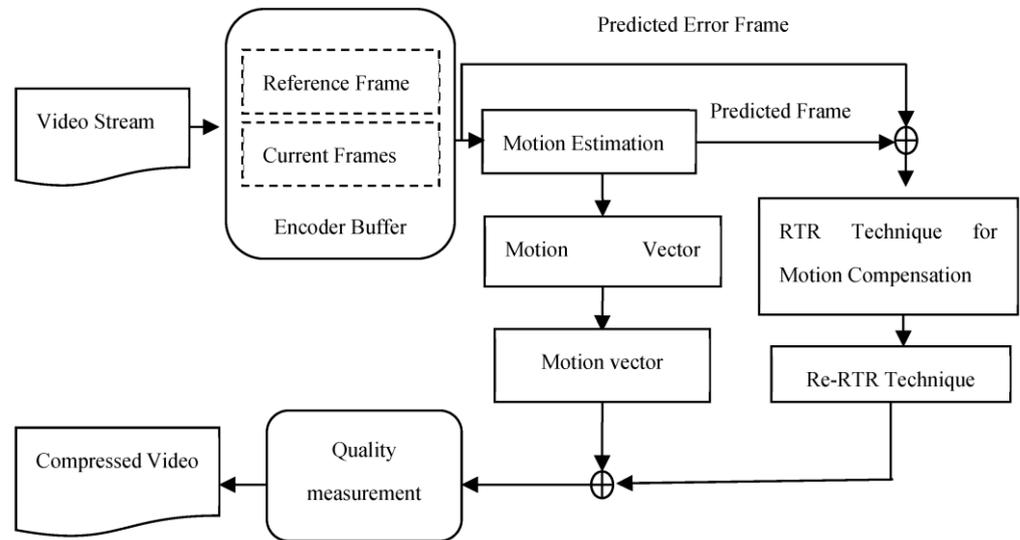


Figure 1. Block diagram of RTR video compression approach.

The previous frames are encoded without data loss so that the decoder decodes $I(n - 1)$ flawlessly. These symmetrical frames are sent to the matching algorithm to find the estimated movement in each block. The idea of the motion vector has been previously elaborated in some simulation algorithms trying three-step-search (NTSS) in [25,26] and four-step search (4SS) in [27]; however, these were all tested on some hypothetical scenarios comparing the results in the form of MSE per pixels. The need for implementation in a real-time scenario was addressed in this paper. Further, our approach appears to be based on the results of these well-established approaches from more than a decade ago and hence is better suited for real-time applications. RTR is compared with contemporary protocols in Table 1.

Table 1. Standard families for video transmission [26].

Application and Standard Family	Multiplex Protocol	Video Coding Standards Used	Data Rate	Packet Size	Error Characteristics
SDN Videophone (H.320)	H.221	H.261 and H.263	64–384 kbit/s	N/A	Error-free
PSTN Videophone (H.324)	H.223	H.263	20 kbit/s	100 bytes	Very few bit errors and packet losses
Mobile Videophone (H.324 wireless)	H.223 w/mobile extensions	H.263	10–300 kbit/s	100 bytes	BER = 10×10^{-3} to 10×10^{-5} , losses of H223 packets
Videophone over Packet Network (H.323)	H.225/RTP/UDP/IP	H.261, H.263, MPEG-2	10–1000 kbit/s	≤ 1500 bytes	BER = 0, 0–30% packet losses
Cable/Satellite TV	H.222	MPEG-2	6–12 Mbit/s	N/A	Almost error-free
Videoconferencing over 'Native' ATM (H.310, H.321)	H.222	MPEG-2	1–12 Mbit/s	53 bytes (ATM cell)	Almost error-free
RTR Technique	RTR protocol	AVI (format)	60.76 Mbit/s (compressed from 67.7 MB to 2.97 MB)	64 bytes	0.01~0.03

2.2. Motion Vectors

Each block of symmetrical frames is represented by a vector that shows the motion within it. Therefore, the encoder sends motion vectors of frame $I(n)$ relative to frame $I(n - 1)$ to prepare the motion-compensated predicted frame $PF(n)$. The selected video shows a total of 345 motion vectors in both the encoder and decoder for all frames.

2.3. Motion Estimation

In the entire motion-based video compression process, motion estimation is the most time-consuming and expensive step in the process. This is because the block-matching technique is more efficient than other motion estimation techniques. Every 16×16 block in a temporally prior frame of the current symmetrical frame and the latter frame of the reference frame is subject to block matching. The destination block is defined as the resultant of the most closely matched frame from the previous frame. The motion vector is defined as the displacement of a block between its original and destination blocks. The motion vector gives a rough indication of how objects move in a sequence between frames.

The spatial domain process is also involved in the block-matching process. As a result, the pixels in the video signal are more correlated. A highly correlated video signal is generated by exploiting temporary redundancy elements in the video signal. In Equation (2), the threshold value is checked to ensure that it is equal to the value that can be used to remove the temporal redundancy from the received frames of the video [21].

$$I(x, y, t) - I(x, y, t + 2) < I(x, y, t) - I(x + 2, y, t) \quad (2)$$

where x and y are the pixel space coordinates, and t is the video instance time.

2.4. Motion Compensation

Video frames can suffer from changes due to camera motion effects, such as zooming, camera panning, and swiveling, rotating, or tilting of the camera. There are several specific algorithms designed to detect and compensate for camera effects in video frames due to changes associated with camera movement. In addition to these changes, there may be changes that occur as a result of the movement of the object, such as translation, rotation, occlusion, uncovering, or morphing. The frame is constructed by motion compensation, where each block is a copy of the block from the previous frame shifted by the computed motion vector. The difference between the compensated and reference frames is calculated from the sequence and is referred to as an error or residual frame. In the residual frame, the signal energy tends to be very small, so compression happens more easily as a result.

2.5. Removing Temporal Redundancy (RTR) Technique

In Figure 2, a block diagram of the RTR procedure is shown. Video sequences contain redundant data in terms of temporal information compared to spatial information. Thus, they are converted into video frames consisting of more spatially redundant data as a result of using the transformation algorithm. The spatially redundant data are efficiently decoded, stored in memory, and the redundant data are discarded. After decoding the stored data with the help of the RTR decoder, the video data are reconstructed. This was based on the proposed RTR technique to perform video compression using a block-matching technique.



Figure 2. Block diagram of the removing temporal redundancy technique.

The technique compares two frame estimates from the current symmetric frame to the reference frame in order to produce a compensated predicted frame (PF). After it is calculated, the PF is sent to the output buffer. This threshold value is used to determine which PFs should be merged with reference frames based on the minimum signal-to-noise ratio among all available PF information, using Equation (3) of Section 2.7.

$$\begin{aligned} & \text{If } (PSNR \times PF) < (PSNR)_{min} \\ & \text{If } \left(10 \log_{10} \left(\frac{(255)^2}{MSE} \right) \times (PF) \right) < \left(10 \log_{10} \left(\frac{(255)^2}{MSE} \right) \right)_{min} \\ & \text{Then } PF \cong PF + \text{reference frame} \\ & \text{else } PF \end{aligned}$$

This assumption is made based on the RTR technique, where pixels with high temporal correlation are used to achieve good image quality. Various quality measurement techniques are used to measure the performance of each RTR technique.

2.6. Techniques for Measuring Quality

Obviously, for an effective comparison of compression techniques, it should be possible to quantify the degradation caused by the compression process on the video frame. Since the perceived quality of the video frame image depends on its use and the viewer, quantifying this perception is a mystery. Further, image quality has many components, and the general problem of producing a cost function that incorporates all of these components is believed to be intractable. Nevertheless, several metrics have been proposed to allow comparisons to be made between compression schemes. In this study, PSNR, MSE, and MAE were used to measure the quality of compressed images [22].

2.7. Peak Signal-to-Noise Ratio (PSNR)

The PSNR measurement indicates how much error is present in the eight-bit PF. The PSNR is a normalized quality metric based on MSE, which is a distortion metric. Accordingly, the normalized PSNR of the PF can be defined as given in Equation (3).

$$PSNR = 10 \log_{10} \left(\frac{(255)^2}{MSE} \right) \text{dB} \quad (3)$$

The mean square error (MSE) of a frame pixel is defined as Equation (4).

$$MSE = \frac{\sum_{x=1}^{X^{SIZE}} \sum_{y=1}^{Y^{SIZE}} (p_{x,y} - \hat{p}_{x,y})^2}{X^{SIZE} \cdot Y^{SIZE}} \quad (4)$$

where X^{SIZE} and Y^{SIZE} are the dimensions of the frame in pixels, $p_{x,y}$ represents the pixel at the x and y direction in the current symmetric frame, and $\hat{p}_{x,y}$ represents the pixel at the x and y direction in the reference frame. The MSE values of current symmetric frames were available for 61 frames, providing an aggregated measure of error for an entire frame, as shown in Figure 3. During the encoding of video signals for video compression, the RTR technique was used to encode the video signals. The MAE is also used to differentiate between two frames based on whether a camera or object within a frame is moving.

A parallel may be drawn with similar results in [25–29], which appeared to be based on a simulation of hypothetical scenarios. Nevertheless, our results are based on compressing the transmission of a real-time video clip and appear to be based on their work, which is cross-validated with contemporary protocols in Table 1.

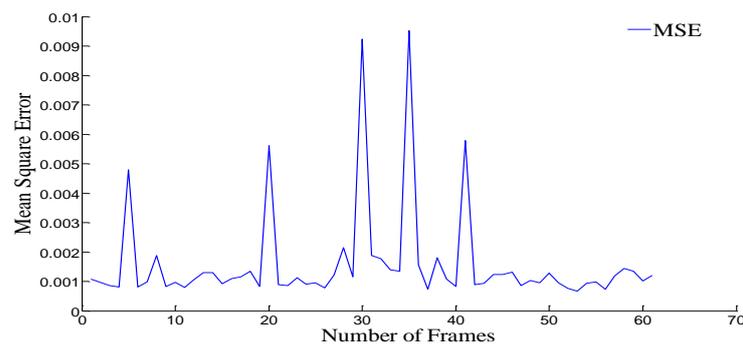


Figure 3. Mean square error of 61 frames.

2.8. Mean Absolute Error (MAE)

Figure 4 shows the MAE values of the first 61 frames. The MAE shows how far predicted values in a given frame are away from observed values in the reference frame and is calculated by subtracting the symmetric current frame from the previous frame. The MAE is obtained by taking the square root of Equation (5).

$$MAE = \frac{\sum_{x=1}^{X^{SIZE}} \sum_{y=1}^{Y^{SIZE}} \sqrt{(p_{x,y} - \hat{p}_{x,y})^2}}{X^{SIZE} \cdot Y^{SIZE}}$$

$$MAE = \frac{\sum_{x=1}^{X^{SIZE}} \sum_{y=1}^{Y^{SIZE}} (p_{x,y} - \hat{p}_{x,y})}{X^{SIZE} \cdot Y^{SIZE}} \quad (5)$$

Now, the absolute of Equation (6) is taken to get the normalized form of MSE.

$$MAE = \frac{1}{X^{SIZE} \cdot Y^{SIZE}} \sum_{x=1}^{X^{SIZE}} \sum_{y=1}^{Y^{SIZE}} \text{abs}(p_{x,y} - \hat{p}_{x,y}) \quad (6)$$

Thus, in a video signal, much of the information that represents a current symmetric frame is the same as the information that represents the reference frame.

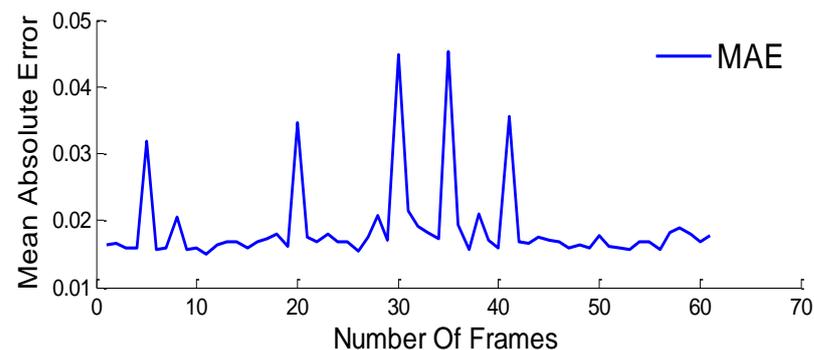


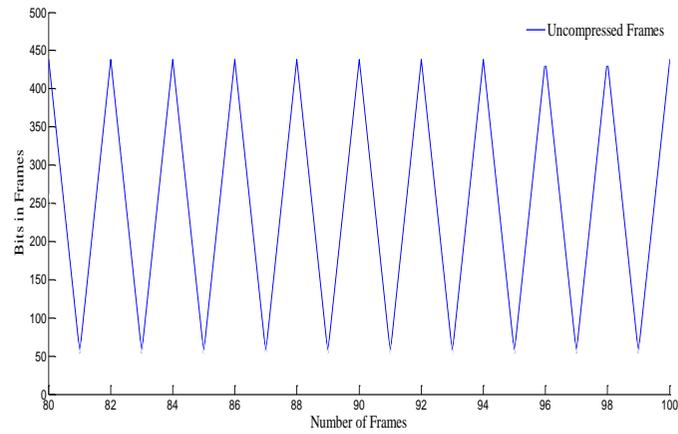
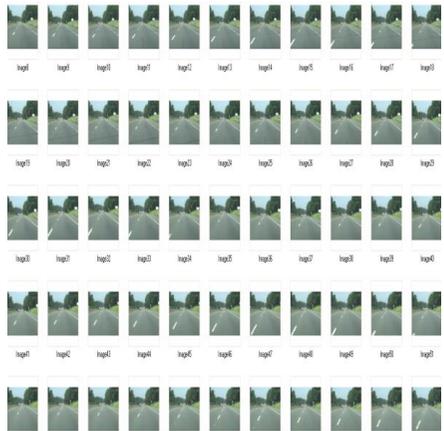
Figure 4. Mean absolute error of the first 61 frames.

3. Results and Discussion

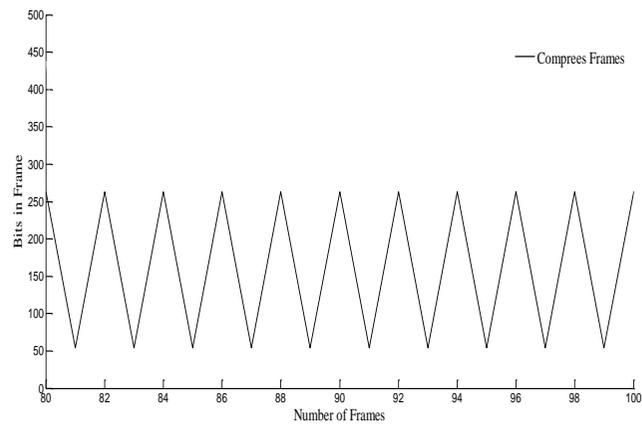
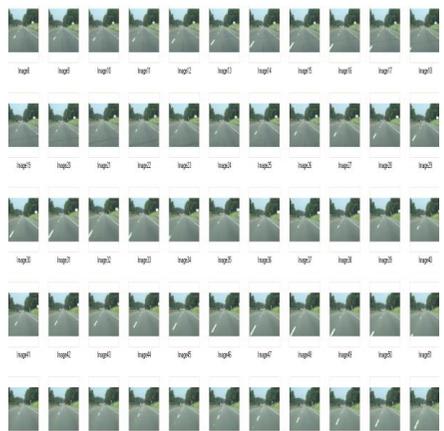
There was the possibility of transmitting a test video signal from the signal source to the signal destination via a real-time wireless communication system. It was a 9 s clip of 67.7 MB in “AVI” format. Video compression is the method by which a video stream is transmitted over a channel with limited bandwidth. Video transmission can sometimes cause fluctuations in the received video signal as a result of transmission. There are several target networks, video coding and multiplexing standards, and transmission characteristics listed in Table 1. These characteristics include things such as packet size and error rates using the RTR technique.

Figure 5 shows a comparison between the original and compressed video signals. The study showed that by using the RTR control algorithm, the bits in the frame became more stable and so more suitable for real-time transmission in the current symmetric network environment. The largest burst was reduced from 450 to 250 kB, which was only half of the original network burst. The compressed video file size was reduced from 67.7 to 2.97 MB, which was 95.61% of the original network burst size shown in Table 1. As a result, the RTR was more suitable for the real-time transmission of compound video sequences to avoid temporal redundancy.

To improve the quality of compressed videos, scientists have relied on the importance of ensuring that the communication network is error resistant [24]. The RTR technique implied that the compressed video was entirely appropriate for mobile communication, as listed in Table 1. The input video frames were separated and converted into grayscale still images in JPEG format. The motion estimation and compensation operations are usually performed using the block-based motion estimation algorithm. The proposed RTR technique achieved motion-compensated frames with a compression ratio that resulted in 95.61% data rate reduction. The PSNR value was obtained from the motion-compensated frame. This technique was also used to obtain the reconstructed image after performing the compression, as shown in Figure 6. There was a significant reduction in the size of the compressed video, which ultimately resulted in a lower data transmission rate.



(a)



(b)

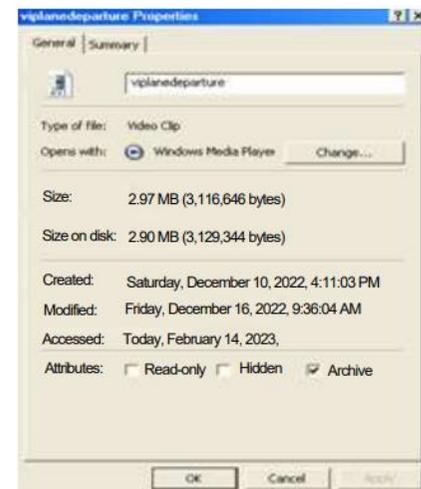
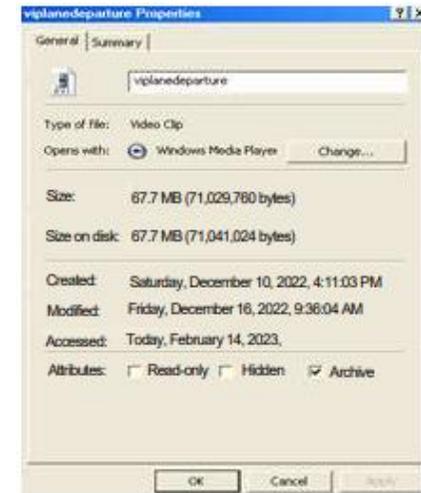


Figure 5. Frames of the uncompressed and compressed video. (a) Original video results; (b) RTR compression results.

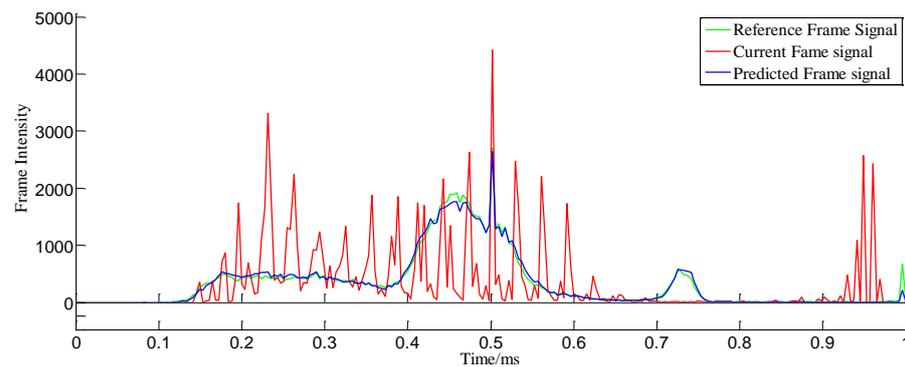


Figure 6. Performance of RTR algorithm at 15.7 PSNR.

Figure 6 shows the performance of the RTR algorithm with different intensities of frames. In this experiment, the target data rates were scalable from 500 to 6000 kbps for the video clip *Viplanedepture.avi* at 30 frames per second frame rate. Scaling the data rate across different data rates and keeping the video quality relatively constant irrespective of the data rate is referred to as block-based motion compensation. With fewer motion-compensating blocks, the PSNR value decreased much more rapidly. At 15.7 dB PSNR, the target frame intensity increased to 4000 kbps, as shown in Figure 6, while this value decreased at an increased PSNR of 23.4 dB and 34.7 dB PSNR, as shown in Figures 7 and 8 respectively. As can be seen from Figure 9, where the differences between the minimum and maximum PSNR values within motion-compensated blocks are illustrated and compared with the PSNR values of the uncompensated frames, the PSNR values increased and decreased. Although standard JPEG compression had a similar effect regarding maximum and minimum PSNR values, the RTR compression technique achieved better results than JPEG compression regarding maximum and minimum PSNR values. From one frame to another, the PSNR curve was constantly changing from 15.7 to 34.7 dB, and the disparity fluctuated from frame to frame. At this point, in addition to increasing the PSNR values, the motion-compensated frame showed a reasonable amount of consistency.

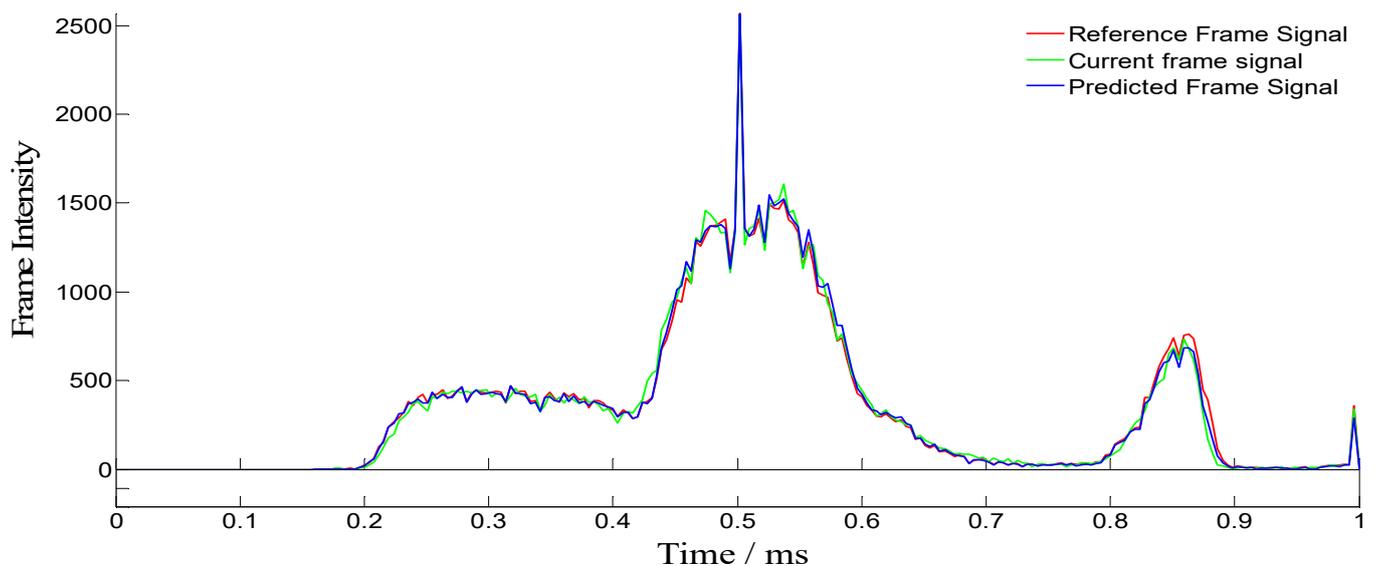


Figure 7. Performance of RTR algorithm at 23.4 dB PSNR.

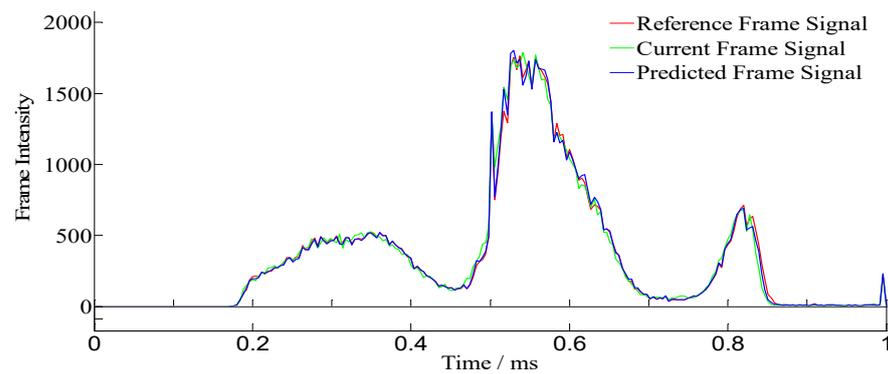


Figure 8. Performance of RTR algorithm at 34.7 dB PSNR.

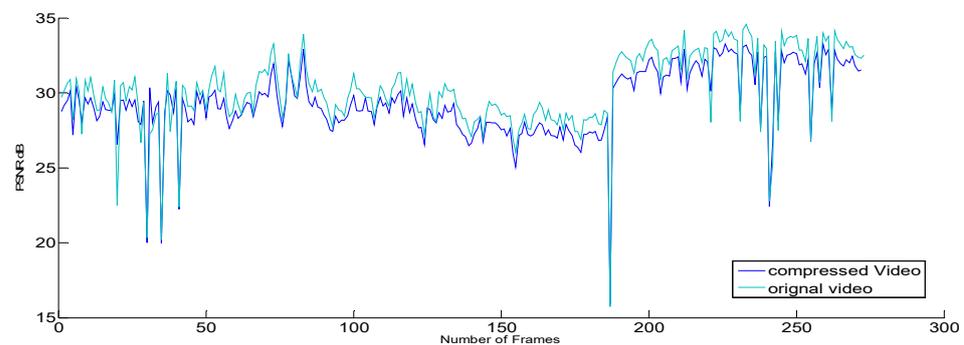


Figure 9. PSNR of original and compressed video JPEG frames.

The RTR technique compressed 67.7 MB (the equivalent of 71,029,760 bytes) to 2.97 MB (the equivalent of 3,116,646 bytes), as shown in Figure 8, with an error of 0.01–0.3 Mbit, which indicated the applicability of RTR in mitigating bandwidth-related issues more effectively. The results were produced through purposefully developed code that directly used data frames, referred to as experimental implementation of RTR. As shown in Table 1, the paper relied on identically produced contemporary algorithms for cross-validation instead of ML-based networks that required training on datasets [26]. The authors believe that the results exhibited similarity with the results of [30], if the latter range of 0.1 to 0.5 bits per pixel was extended to the 0–250 frames range. As shown in Table 1, the authors proposed the use of an algorithm that selected AVI files and compared them with a variety of contemporary standards instead of AVC (H264), HEVC (H265), and VVC (H266), which are already established techniques using the database.

The PSNR values of the original and compressed video JPEG frames in Figure 8 attested to the utility of the method by so closely resembling the predicted frame signal of Figure 9 using the RTR technique at 34.7 dB PSNR, which was comparable to 44 dB tested under more demanding conditions. The trend in the PSNR graphs was comparable to that obtained by the authors in [31,32], where an improved method of embedding a watermark in a high-quality video based on DNN under high-efficiency video coding (HEVC) compression conditions was presented. The graphs in this work showed an upward trend beyond epoch 180, contrary to that in [33], indicating a reduction in the PSNR values for epochs beyond 200.

Evaluation of the proposed method was simulated on grayscale images taken from real-time video, including road images, as shown in Figure 10. The test image was 8 bits/pixel, and the reduction scheme was tested on images of 256×256 and 512×512 in size. The largest block sizes for 512×512 and 256×256 images were 32×32 and 16×16 pixels, respectively. The simulation platform was Microsoft Windows XP, Pentium III, and the suggested approach was implemented using MATLAB. The performance of the reduction scheme was evaluated.

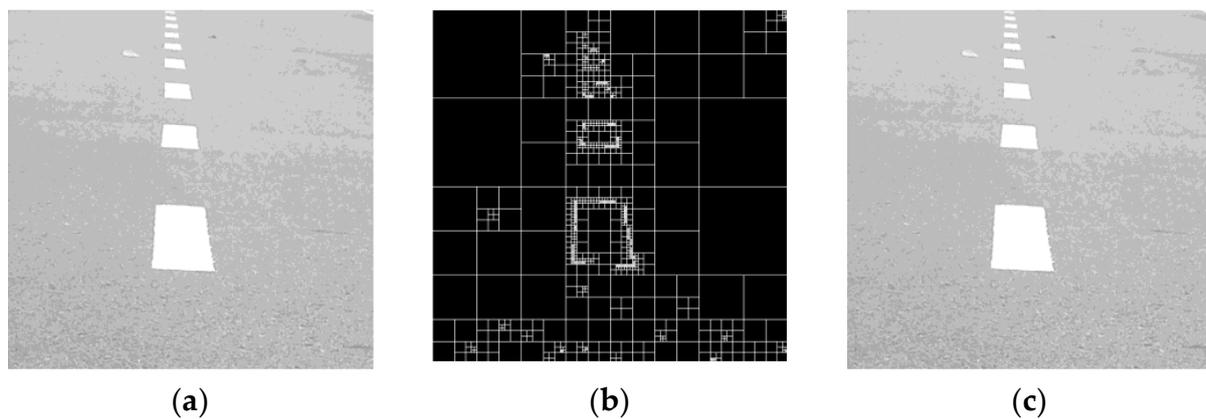


Figure 10. Evolution performance of decomposition and shuffling of an image. (a) Original image; (b) redundancy image; (c) compressed image.

A reduction scheme is used to increase internal space and bandwidth in order to increase the communication speed. The compression process was completed for different image block sizes. Figure 10 shows how the image blocks were maximized when the image was decomposed and shuffling was performed. For varying image sizes, the results indicated similarity in the images.

4. Conclusions

The issue of redundancy in video frames was addressed against the current symmetric video compression techniques that have not adequately reduced this redundancy. In this paper, a video compression method called RTR was proposed, which took into account the temporal redundancy of the video without bias. This method offered the best compromise between quality and complexity at high bit rates. It provided an acceptable level of performance in almost all possible bit rate variants. It was observed that the compression method based on RTR could produce a better compression ratio as well as a better PSNR value than JPEG video compression. There are various directions for the future exploration of the 2675 kbps data rate, such as combining it with transformations to reduce the computational complexity of the video compression process. As a type of compression technique, block-based motion estimation and compensation can be a suitable option for wireless mobile networks since they offer a convincing compression ratio and are relatively simple in their architecture. The RTR technique could be implemented without any transformation error or with minor characteristic error (0.01~0.03), which wastes the largest percentage of energy within the compression scheme, even though RTR techniques for video compression encoders are encouraged in wireless mobile network applications. The paper claims a reasonable consistency of the RTR technique, showing PSNR values varying from 15.7 to 34.7 dB. The proposed algorithm selects AVI files and compares them with a variety of contemporary standards instead of AVC (H264), HEVC (H265) and VVC (H266), which are already established techniques using the database.

Author Contributions: Conceptualization, M.I. and S.H.; Methodology, S.H.; Software, S.H. and M.I.; Validation, M.F.A. and M.I.; Formal analysis, S.H., H.I.S. and M.M.S.; Investigation, W.A.; Resources, W.A. and M.F.A.; Data curation, M.I.; Writing—original draft preparation, S.H. and M.I.; Writing—review and editing, M.M.S., W.A., H.I.S. and M.F.A.; visualization, M.M.S., H.I.S., W.A. and M.F.A.; Supervision, S.H.; Project administration, M.F.A.; Funding acquisition, H.I.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education, Saudi Arabia, for funding this research work through the project number (QU-IF-2-4-5-26275). The authors also thank Qassim University for technical support.

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

References

1. Abdhussain, S.H.; Al-Haddad, S.A.R.; Saripan, M.I.; Mahmmod, B.M.; Hussien, A. Fast Temporal Video Segmentation Based on Krawtchouk-Tchebichef Moments. *IEEE Access* **2020**, *8*, 72347–72359. [[CrossRef](#)]
2. Soh, J.W.; Park, J.; Kim, Y.; Ahn, B.; Lee, H.-S.; Moon, Y.-S.; Cho, N.I. Reduction of Video Compression Artifacts Based on Deep Temporal Networks. *IEEE Access* **2018**, *6*, 63094–63106. [[CrossRef](#)]
3. Battiato, S.; Bruna, A.R.; Puglisi, G. A Robust Block-Based Image/Video Registration Approach for Mobile Imaging Devices. *IEEE Trans. Multimed.* **2010**, *12*, 622–635. [[CrossRef](#)]
4. Xiao, J.; Li, S.; Xu, Q. Video-Based Evidence Analysis and Extraction in Digital Forensic Investigation. *IEEE Access* **2019**, *7*, 55432–55442. [[CrossRef](#)]
5. Xun, C.; Lim, J.S. Algorithms for Transform Selection in Multiple-Transform Video Compression. *IEEE Trans. Image Process.* **2013**, *12*, 5395–5407.
6. Yeongyoon, C.; Nosratinia, A. Analysis of three-phase multiple access with continual contention resolution (TPMA-CCR) for wireless multi-hop ad hoc networks. *J. Commun. Netw.* **2011**, *13*, 43–49.
7. Yue, M.C.; Bajic, I.V.; Saedi, P. Moving Region Segmentation From Compressed Video Using Global Motion Estimation and Markov Random Fields. *IEEE Trans. Multimedia* **2011**, *13*, 421–431.
8. Gonçalves, D.D.O.; Costa, D.G. A Survey of Image Security in Wireless Sensor Networks. *J. Imaging* **2015**, *1*, 4–30. [[CrossRef](#)]
9. Bosch, M.; Fengqing, Z.; Delp, E.J. Segmentation-Based Video Compression Using Texture and Motion Models. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 1366–1377. [[CrossRef](#)]
10. Yang, X.; Yan, Z.; Nolen, M.; Deng, J.H.; Jingyuan, Z. A Cross-Layer Approach for Prioritized Frame Transmissions of MPEG-4 Over the IEEE 802.11 and IEEE 802.11e Wireless Local Area Networks. *IEEE Syst. J.* **2011**, *5*, 474–485.
11. Lai, M.P.; Ka, H.N.; Ka, M.W.; Kwok, W.C. Multi Direction Search Algorithm for Block-Based Motion Estimation. In Proceedings of the APCCAS 2008-2008 IEEE Asia Pacific Conference on Circuits and Systems, Macao, 30 November 2008–3 December 2008.
12. Habib, S.; Alsanea, M.; Aloraini, M.; Al-Rawashdeh, H.S.; Islam, M.; Khan, S. An Efficient and Effective Deep Learning-Based Model for Real-Time Face Mask Detection. *Sensors* **2022**, *22*, 2602. [[CrossRef](#)]
13. Aziz, T.; Dolly, D.R.J. Motion Estimation and Motion Compensated Video Compression Using DCT and DWT. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 667–671.
14. Habib, S.; Hussain, A.; Albattah, W.; Islam, M.; Khan, S.; Khan, R.U.; Khan, K. Abnormal Activity Recognition from Surveillance Videos Using Convolutional Neural Network. *Sensors* **2021**, *21*, 8291. [[CrossRef](#)]
15. Zhou, W.; Duan, Z.; Hongqi, H. Fast motion estimation algorithm for H.264/AVC based on centered prediction. *J. Syst. Eng. Electron.* **2010**, *21*, 1103–1110. [[CrossRef](#)]
16. Roman, S.; Radim, H.; Zdenek, M. Compression methods for image processing implementation into the low capacity devices. *Teh. Vjesn. Tech. Gazette.* **2013**, *20*, 1087–1090.
17. Jigar, R.; Rutika, J.; Manish, S. Review Towards Fast Block Matching Algorithm for Video Motion Estimation. In Proceedings of the International Conference on Communication Systems and Network Technologies, Indore, India, 23–24 April 2012.
18. Wu, Y.G.; Huang, G.F. Motion vector generation for video coding by gray prediction. *IET Comput. Vis.* **2011**, *5*, 14–22. [[CrossRef](#)]
19. Purnachand, N.; Luis, N.A.; Antonio, N. Improvements To TZ Search Motion Estimation Algorithm For Multi view Video Coding. In Proceedings of the International conference on systems, signals and image processing (IWSSIP), Vienna, Austria, 11–13 April 2012; pp. 388–391.
20. Han, H.; Woods, J.W.; Yao, Z.; Huihui, B. Control-Point Representation and Differential Coding Affine Motion Compensation. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1651–1660.
21. Ruiz, G.; Michell, J.; Buron, A. Parallel-pipeline 8/spl times/8 forward 2-D ICT processor chip for image coding. *IEEE Trans. Signal Process.* **2005**, *53*, 714–723. [[CrossRef](#)]
22. Manoranjan, P.; Manzur, M. Video Coding Focusing on Block Partitioning and Occlusion. *IEEE Trans. Image Process.* **2010**, *19*, 691–701.
23. Alsharekh, M.F.; Islam, M.; Ibrahim, A.H.; Khan, R.; Habib, S. Bit Error Rate Performance of RFID Signal in SDR Communication. *J. Appl. Sci.* **2016**, *16*, 161–166. [[CrossRef](#)]
24. Landalufe, H.; Arjona, L.; Perallos, A.; Falcone, F.; Angulo, I.; Muralter, F. A Review of IoT Sensing Applications and Challenges Using RFID and Wireless Sensor Networks. *Sensors* **2020**, *20*, 2495. [[CrossRef](#)]
25. Hyo, S.K.; Jong, H.L.; Chang, K.K.; Byung, G.K. Zoom Motion Estimation Using Block-Based Fast Local Area Scaling. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1280–1291.
26. Yao, W.; Stephan, W.; Jiangtao, W.; Katsaggelos, A.K. Review of Error Resilient Coding Techniques for Real-Time Video Communications. *IEEE Signal Process. Mag.* **2000**, *17*, 61–82.

27. Li, R.; Zeng, B.; Liou, M.L. A new three-step search algorithm for block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* **1994**, *4*, 438–442.
28. Liu, L.-K.; Feig, E. A block-based gradient descent search algorithm for block motion estimation in video coding. *IEEE Trans. Circuits Syst. Video Technol.* **1996**, *6*, 419–422.
29. Zhu, S.; Ma, K.K. A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. Image Process.* **2000**, *9*, 287–290. [[CrossRef](#)]
30. Cheung, C.-H.; Po, L.-M. A novel cross-diamond search algorithm for fast block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.* **2002**, *12*, 1168–1177. [[CrossRef](#)]
31. Zhu, S.; Tian, J.; Shen, X.; Belloulata, K. A new cross-diamond search algorithm for fast block motion estimation. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; IEEE: Piscataway, NJ, USA; pp. 1581–1584.
32. Rodrigues, R.; Grilo, A.; Santos, M.; Nunes, M.S. Native ATM Videoconferencing based on H. 323. In Proceedings of the II Conference on Telecommunications, ConfTele, Sesimbra, Portugal, April 1999; Volume 99.
33. Kaczyński, M.; Piotrowski, Z.; Pietrow, D. High-Quality Video Watermarking Based on Deep Neural Networks for Video with HEVC Compression. *Sensors* **2022**, *22*, 7552. [[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.