Hindawi
Journal of Mathematics
Volume 2024, Article ID 9782496, 1 page
https://doi.org/10.1155/2024/9782496

Retraction

Retracted: Research on Fast Compensation Algorithm for Interframe Motion of Multimedia Video Based on Manhattan Distance

Journal of Mathematics

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] N. Li and S. Wan, “Research on Fast Compensation Algorithm for Interframe Motion of Multimedia Video Based on Manhattan Distance,” *Journal of Mathematics*, vol. 2022, Article ID 3468475, 10 pages, 2022.

Hindawi
Journal of Mathematics
Volume 2022, Article ID 3468475, 10 pages
https://doi.org/10.1155/2022/3468475

Research Article

Research on Fast Compensation Algorithm for Interframe Motion of Multimedia Video Based on Manhattan Distance

Ning Li and Shuai Wan

School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Ning Li; ning_li@mail.nwpu.edu.cn

Received 18 November 2021; Revised 6 December 2021; Accepted 10 December 2021; Published 5 January 2022

Academic Editor: Naeem Jan

To improve the video quality, aiming at the problems of low peak signal-to-noise ratio, poor visual effect, and low bit rate of traditional methods, this paper proposes a fast compensation algorithm for the interframe motion of multimedia video based on Manhattan distance. The absolute median difference based on wavelet transform is used to estimate the multimedia video noise. According to the Gaussian noise variance estimation result, the active noise mixing forensics algorithm is used to preprocess the original video for noise mixing, and the fuzzy C-means clustering method is used to smoothly process the noisy multimedia video and obtain significant information from the multimedia video. The block-based motion idea is to divide each frame of the video sequence into nonoverlapping macroblocks, find the best position of the block corresponding to the current frame in the reference frame according to the specific search range and specific rules, and obtain the relative Manhattan distance between the current frame and the background of multimedia video using the Manhattan distance calculation formula. Then, the motion between the multimedia video frames is compensated. The experimental results show that the algorithm in this paper has a high peak signal-to-noise ratio and a high bit rate, which effectively improves the visual effect of the video.

1. Introduction

In recent years, with the rapid development of multimedia and network technology, video, image, computer vision, multimedia database, and computer network technology are increasingly integrated, covering all aspects of the national economy and social life. Video processing, video coding, and video communication, which are in the core position, have become the frontier fields and hot topics of information and communication engineering [1]. Among them, with the gradual penetration of multimedia, the boundaries among videos, graphics, computer vision, multimedia database, and computer network become blurred, making video processing a multi-disciplinary research field [2]. At present, video processing has been at the core of multimedia technology [3]. At the same time, with the rapid development of video technology, the research of motion compensation between multimedia video frames is very important and necessary. Therefore, video interframe motion compensation plays an important role in many technologies included in video processing [4].

As one of the classic problems in the field of image and video processing, video interframe motion compensation is widely used in video frame rate improvement, slow video production, and virtual view synthesis. At present, the commonly used interframe motion compensation method of the video image is to intensively match the input image pairs based on the optical flow field estimation algorithm and interpolate the input image pixel by pixel using the obtained dense matching information to synthesize the intermediate frame image. As the optical flow field estimation itself is an ill-conditioned problem, especially in the case of weak image texture or occlusion, the effect is poor, and the peak signal-to-noise ratio is low. The existing methods often face difficulties in practical application. In recent years, the method based on deep learning has attracted extensive attention and has achieved remarkable results in many computer vision problems, such as target classification, face recognition, and so on. However, the key to the success of this method is to use a large number of training samples to train the appropriate depth neural network, which has the problems of long compensation time and low bit rate. In addition, reference [5] proposes a deep convolution neural network algorithm to realize image interframe compensation. Firstly, the image interframe compensation model is constructed according to the deep convolution neural network. Secondly, the image features of the compensation model are extracted by sparse self-coding and linear decoding. Then, the image features are mapped by a multilayer convolution neural network. Finally, the image frame resolution is reconstructed according to the sparse algorithm to compensate for the image frame. The experimental results show that the image frame compensation method based on a deep convolution neural network can effectively solve the problem of image loss, however, the bit rate is low and the application effect is poor.

To improve the peak signal-to-noise ratio, the visual effect, and the bit rate of multimedia video images, a fast interframe motion compensation algorithm based on the Manhattan distance is proposed in this paper. Section 2 of this paper presents the processing of the multimedia video. In Section 3, the multimedia video interframe motion fast compensation algorithm is presented. Section 4 proposes the simulation experiments that verify the strength of our method, and the conclusion is given in Section 5.

2. Multimedia Video Preprocessing

2.1. Gaussian Noise Variance Estimation. For the noise level mixed in the multimedia video, this paper adopts the variance to measure. Use the absolute median difference based on the wavelet transform [6] to estimate the Gaussian noise standard deviation of a noisy multimedia video A:

$$\widehat{\partial} = \mathrm{MAD}\left(x_a\right) * \left(1 - \frac{S_y}{S}\right). \tag{1}$$

x_a represents the first-level fine-scale wavelet coefficients of the multimedia video A. The MAD operator is defined as follows:

$$\mathrm{MAD}\left(x_a\right) = \frac{\mathrm{Med}\left(E_1\right)}{K_w W_1}. \tag{2}$$

$\mathrm{Med}(\cdot)$ represents the median of the input vector. The fast wavelet transform ensures the high execution rate of the MAD operator, making it suitable for the batch estimation of the Gaussian noise variance of video frames [7]. The specific Gaussian noise variance estimation formula is as follows:

$$\sigma_a^2 = \frac{I_r \sqrt{\left(R_r/\widehat{\partial}\right)^2}}{\omega_s}. \tag{3}$$

I_r represents the high frequency information of the video. R_r represents the low frequency information of the video. ω_s represents the amount of noise in the video.

2.2. Active Noise Mixing Forensics Algorithm. According to the results of the Gaussian noise variance estimation, this paper proposes an active noise mixing forensics algorithm. Firstly, preprocess the original video for noise mixing, i.e., use a pseudo-random sequence to generate Gaussian noise with a standard deviation of s^2 and add it to each pixel of the video sequence. Then, the processed video will be tampered with by Frame Rate Up-Conversion (MC-FRUC) to generate an up-converted video, and attacks such as denoising and compression may also be implemented. Finally, analyze the Gaussian noise distribution of the suspicious video to identify whether there is MC-FRUC tampering. The following specifically introduces the core of the proposed algorithm: noise mixing, forensics, and detection.

Assume that the original video sequence is composed of K video frames of size $n \times m$. The pseudo-random sequence can be used to generate 0-mean Gaussian noise with a standard deviation of s^2, and the pixels of the original video sequence are added as

$$h_k(i, j) = e_k(i, j) + l_k(i, j). \tag{4}$$

$e_k(i, j)$ and $h_k(i, j)$, respectively, represent the original frame of the k^{th} frame and the pixel value of the noisy frame at position (i, j). $l_k(i, j)$ represents the value of the mixed Gaussian noise. When the original video sequence encounters MC-FRUC tampering, the combination of the two original frames r_{k-1} and r_{k+1} must insert the current frame $\widehat{r_k}$, and formula (4) is derived to obtain

$$\widehat{h_k}(i, j) = \widehat{e_k}(i, j) + \widehat{l_k}(i, j). \tag{5}$$

Of which,

$$\widehat{e_k}(i, j) = \prod_{i=1}^{N} \prod_{j=1}^{N} P_{ij}\left(1 - P_\alpha\right), \tag{6}$$

$$\widehat{l_k}(i, j) = \prod_{i=1}^{N} \prod_{j=1}^{N} \left(U_{ij} \times \ln\left(1 - P_\beta\right)\right). \tag{7}$$

P_{ij} represents the adjacent frames. P_α and P_β represent the threshold. U_{ij} represents the corner points of the video frame.

It is observed from formula (7) that the interpolated frame noise term $\widehat{l_k}$ is obtained by the weighted summation of the noise term l_{k-1} of the $k-1^{\text{th}}$ frame and the noise term l_{k+1} of the $k+1^{\text{th}}$ frame along the motion trajectory. Since the components of the noise term are independent of each other, the variance of both ends of formula (5) can be obtained at the same time.

$$\sigma_k^2 = W\left[\widehat{l_k}(i, j)\right] = W\left[C_{k,1}^{(i,j)}, C_{k,2}^{(i,j)}\right]. \tag{8}$$

Since each pixel in h_{k-1} and h_{k+1} is premixed with 0-mean Gaussian noise with a standard deviation of s^2, it can be seen that

$$s^2 = W_{k-1}\left[C_{k,1}^{(i,j)}, C_{k,2}^{(i,j)}\right] \times W_{k+1}\left[C_{k,1}^{(i,j)}, C_{k,2}^{(i,j)}\right]. \tag{9}$$

Substituting formula (9) into formula (8), we can get

$$\sigma_k^2 = \frac{s^2}{2}. \tag{10}$$

According to formula (10), it can be known that the variance σ_k^2 of the interpolated frame $\widehat{l_k}$ is half of the variance s^2 of the mixed noise. MC-FRUC tampering will periodically insert the interpolated frames. Therefore, the noise standard deviation of the fake video will show periodic sudden changes, as shown in Figure 1 (the unforged video is the original 30 fps video, and the fake video is the 15 fps original video up to 30 fps). The standard deviation is premixed into the video Gaussian noise of 5. Using the MAD method to estimate the noise standard deviation of the unforged and forged videos, it can be seen that the noise standard deviation curve of the unforged video changes smoothly and slowly, while the noise standard deviation curve of the forged video changes rapidly and periodically. It can be seen that the periodicity of the noise standard deviation curve can be used as a strong piece of evidence to discriminate the tampering of MC-FRUC.

2.3. Smooth Processing of Noisy Multimedia Video. To improve the accuracy of motion compensation between multimedia video frames, the fuzzy C-means clustering method [8] is used to smoothly process noisy multimedia videos. Use the gray-scale cluster membership matrix to transform the membership tensor in the noisy multimedia video. The detailed process is as follows:

Step1: The determination of the iteration error, the maximum number of iterations, and the number of clustering categories is followed by the obtaining of a segmented initial membership degree θ_z, for which the calculation formula is as follows:

$$\theta_z = \sum_{j=1}^{N} \frac{U_j - P_j}{1 - P_j}. \tag{11}$$

U_j represents the fuzzy compactness function of the global interval value. P_j represents the fuzzy mean value of the local interval.

Step 2: Calculate the label value of the noisy frame in the noisy multimedia video according to the degree of membership θ_z.

$$G(\theta_z) = Q_z - \sum_{j=1}^{N} u_j. \tag{12}$$

Step 3: Perform step t iterative processing based on the tag value, and update the membership tag value.

Step 4: For the new label value of the membership degree, mark according to the principle of maximum membership degree.

Step 5: Obtain the iteration error according to the maximum membership degree of the mark. The calculation formula is as follows:

$$\varepsilon^2 = \left|\theta_{z(t+1)} - \theta_{zt}\right|^2. \tag{13}$$

$\theta_{z(t+1)}$ represents the maximum membership error produced by the $t+1^{\text{th}}$ iteration. θ_{zt} represents the maximum membership error produced by the t^{th} iteration.

On this basis, the obtained gray-scale cluster membership degree is converted into the membership degree tensor corresponding to the multimedia video. After the membership degree tensor is subjected to mean filtering processing, the label value is obtained, and the smooth processing result of the noisy multimedia video can be realized. Figure 2 is a flow chart of the smoothing processing of noisy multimedia videos.

2.4. Extraction of Significant Information from Multimedia Video. It can be seen from the characteristics of the human visual system that the human eye usually notices first the target or the area of interest in the scene, while the remaining noninteresting parts or repetitive content are easily overlooked. The saliency map model is a selective attention model that simulates the visual attention mechanism of organisms. In this paper, the residual spectrum method is used to extract the visual saliency map of the multimedia video [9], which can be expressed as follows:

$$X(d) = Y(d) - Z(d). \tag{14}$$

$Y(d)$ represents the original video logarithmic amplitude spectrum. $Z(d)$ represents the general logarithmic amplitude spectrum after mean filtering. $X(d)$ represents the remaining spectrum. The following is a detailed analysis of the significant information extraction steps of multimedia videos:

Step 1: Perform a two-dimensional Fourier transform on the multimedia video image.

Step 2: Obtain the amplitude spectrum by calculating the absolute value of the transformed video image. Also, calculate the phase spectrum.

Step 3: Obtain the difference amplitude spectrum by subtracting the filtered amplitude spectrum from the amplitude spectrum of the original video image.

Step 4: Reconstruct the video image by two-dimensional inverse Fourier transform using the difference amplitude spectrum and phase spectrum.

Step 5: Obtain the saliency map of the video image by performing the Gaussian filtering and normalization on the reconstructed video image.

Compared with the high-frequency information of the video image, the saliency map of the video image can not only reflect the details of the video image but also extract the

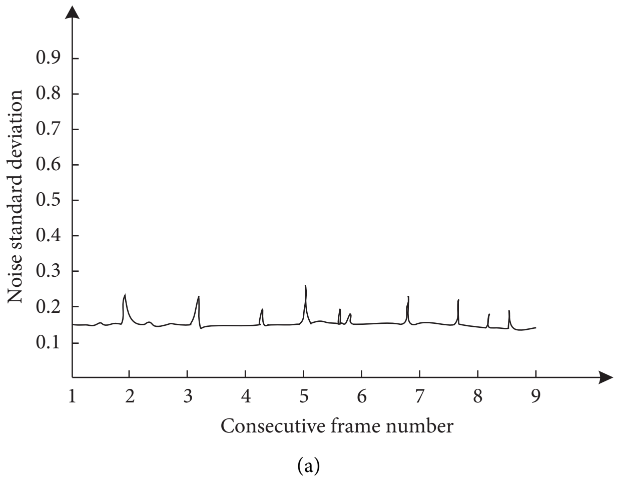


Figure 1: Standard deviation spectrum of unforged noise and forged noise. (a) Unforged noise. (b) Forged noise.

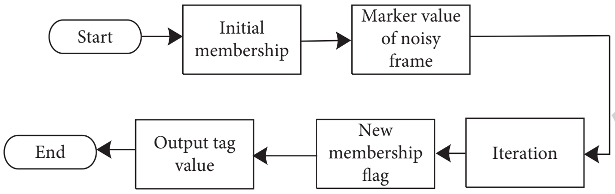


Figure 2: Flow chart of the smoothing processing of a noisy multimedia video.

areas that can attract the attention of the human eye. Therefore, using the saliency map of the video image to represent the details of the video image is more in line with the visual characteristics of the human eye.

3. Multimedia Video Interframe Motion Fast Compensation Algorithm

In the multimedia video preprocessing link, the smooth processing of noisy multimedia video is realized, and the significant information of multimedia video is obtained, which provides a stable basic condition for the fast compensation of multimedia video interframe motion. Next, the fast compensation processing of multimedia video interframe motion will be carried out.

3.1. Multimedia Video Motion Estimation. Extracting object motion information from video images is called motion estimation. The basic principle of general motion estimation is as follows: assume that the video frame $g(x, y)$ at time T represents the current frame, and the video frame $g'(x, y)$ at time T' represents the reference frame. When the reference frame is the previous frame of the current frame, i.e., when $T' = T + \Delta T$, it is called backward motion estimation [10]. When the best position of the block in the current frame $g(x, y)$ is searched in the reference frame $g'(x, y)$, the corresponding motion field $M_{(T,T+\Delta T)}$ can be obtained to obtain the motion vector of the current frame as shown in Figure 3.

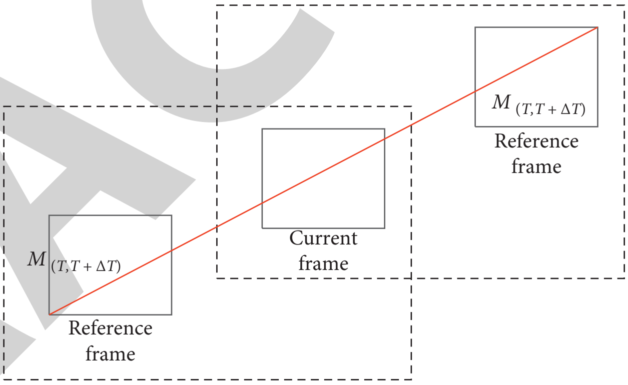


Figure 3: Schematic diagram of motion estimation.

Motion estimation generally adopts block-based motion estimation. The basic idea of block-based motion is to divide each frame of the video sequence into nonoverlapping macroblocks and find the best position of the block corresponding to the current frame in the reference frame according to the specific search range and specific rules, i.e., find the matching block. The relative displacement between the matching block and the current block is the motion vector [11].

The whole implementation process of block matching motion estimation is to find the most matched motion vector, and it reduces the time redundancy of motion compensation by eliminating the time correlation between the current frame and the reference frame. The basic principle of block matching motion estimation is to take the prediction unit as the basic unit, find the corresponding prediction unit in the reference frame for each prediction unit of the current frame in a certain order, and determine the relative displacement between them using the found prediction unit, i.e., the search of the motion vector is completed. The importance of the motion vector cannot be ignored. The more accurate the motion vector prediction, the better the effect of motion estimation.

The ultimate goal of motion estimation is to transmit the motion vector and prediction error to the video decoding end. Motion compensation is to subtract the prediction unit from the current prediction unit based on motion estimation to obtain the residual unit. Such residual unit contains little information. Carry out quantization transformation and entropy coding to obtain the code stream. Therefore, the accuracy of motion estimation directly affects the effect of fast motion compensation between multimedia video frames [12].

Figure 4 shows the basic process of motion estimation. In the figure, time T corresponds to the g^{th} frame image, and time $T-1$ corresponds to the $g-1^{\text{th}}$ frame image. In the $g-1$ frame, find a part that most closely matches the g frame. It is called searching for the best block, and it is judged that the position of the matching block in the $g-1$ frame is the previous position of the block in the g frame, and the displacement of this movement is called the motion vector.

3.2. Realization of Fast Motion Compensation between Multimedia Video Frames

3.2.1. Manhattan Distance. Manhattan distance refers to the distance between the two points strictly based on a horizontal or vertical path, rather than a diagonal line or a distance similar to a straight broken line [13]. It is a simple superposition of the distance between the horizontal and vertical components.

M-dimensional space is a set of points. Each point e of it can be expressed as $\{e(1), e(2), \ldots, e(n)\}$, where $e(i)$ is called the i^{th} coordinate of e, and $i = 1, 2, \ldots, n$. The Manhattan distance between two points $P = \{p(1), p(2), \ldots, p(n)\}$ and $Q = \{q(1), q(2), \ldots, q(n)\}$ can be expressed as

$$D_{pq} = \sum_{i=1}^{N} \| p(i) - q(i) \|^2. \tag{15}$$

The Manhattan distance is applied to the video image. The background image is also the background template. The Manhattan distance is the sum of the Manhattan distances of each pixel:

$$\text{MDB} = \sum_{i=1}^{n} \sum_{j=1}^{n} P_{ij}. \tag{16}$$

The Manhattan distance between the background template and the current frame (MDFB) is the sum of the relative distances between the background image and the corresponding pixels of the current frame:

$$\text{MDFB} = \sum_{i=1}^{n} \sum_{j=1}^{n} P_{ij} - \text{Q}_{ij}. \tag{17}$$

The calculation formula of the relative Manhattan distance RMD between the background template and the current frame to the background is,

$$\text{RMD} = \frac{\text{MDB}}{\text{MDFB}}. \tag{18}$$

In the above formula, MDB represents the Manhattan distance of the background template itself. MDFB represents the Manhattan distance between the background template and the current frame. RMD represents the relative Manhattan distance of the current frame to the background.

3.2.2. Fast Compensation of Motion between Multimedia Video Frames. The Manhattan distance calculation formula is used to obtain the relative Manhattan distance between the current frame of the multimedia video and the background, and the motion between the multimedia video frames is compensated.

(1) Basic Idea. Firstly, judge whether the current frame is a background frame or a target frame according to RMD. Secondly, if it is a background frame, i.e., RMD < TH1, then the current frame is used as the background image. Thirdly, if the change is small, i.e., TH1 ≤ RMD ≤ TH2, the median operation is performed with the adjacent four frame images. The fourth point is that if TH2 ≤ RMD ≤ TH3 is satisfied, then the target frame is subjected to the median operation. The median operation is carried out with the background frame, and the final result is taken as the new background frame. The fifth point is that if there is a large difference, i.e., RMD ≥ TH3, then the target frame is judged as the object starts moving from rest, and the current frame is used as the background frame. The specific implementation formulas are shown in formulas (19) and (20).

$$\begin{cases} \text{RMD} < \text{TH1}, \\ \text{TH1} \leq \text{RMD} \leq \text{TH2}, \\ \text{TH2} \leq \text{RMD} \leq \text{TH3}, \\ \text{RMD} \geq \text{TH3}, \end{cases} \tag{19}$$

$$V(t) = \{v(t), v(t-1), v(t-2), v(t-3), v(t-4)\}. \tag{20}$$

In the above formula, $V(t)$ represents the background model at time t. $v(t)$ represents the current target frame. $v(t-1)$, $v(t-2)$, $v(t-3)$, and $v(t-4)$ are the first four frames immediately adjacent to $v(t)$.

Since the motion of different pixels in the current multimedia video is related to the motion of the candidate block, the motion compensation prediction in the merge mode is not accurate enough. To make full use of the motion correlation between pixels as the distance changes, this section proposes a weighted prediction based on Manhattan distance as an additional candidate for the merge mode [14]. The specific steps are as follows:

(i) Detecting neighbor merge candidate blocks: firstly, the neighbor merge candidate block is detected. Figure 5 is a schematic diagram of the position of the merge mode candidate block.

As shown in the position of the candidate block in the merge mode in Figure 5, the candidate blocks in different positions are detected in the order of α_1, α_2,

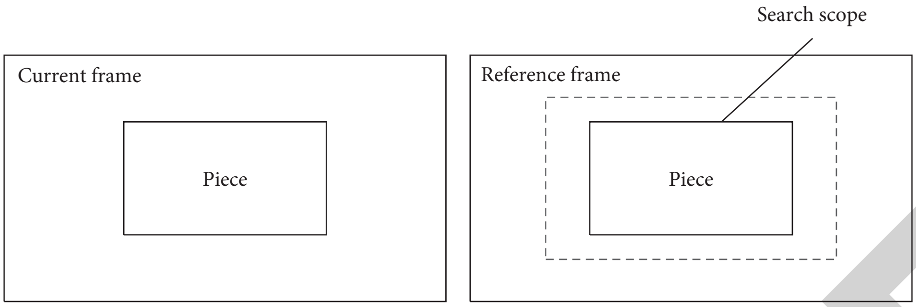


Figure 4: Schematic diagram of motion estimation.

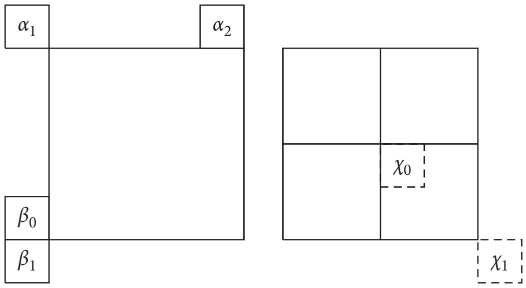


Figure 5: Schematic diagram of the position of candidate blocks in the merge mode.

β_0, χ_0, β_1, and χ_1, and the corresponding motion vector is generated. If the number of generated motion vectors is less than 2, then the algorithm of this paper is not executed. Otherwise, the algorithm of this paper is executed.

(ii) Motion compensation: use the motion vector generated in Step 1 to perform motion compensation prediction to obtain the corresponding prediction block, which is denoted as $P_k(x, y)$. In the algorithm proposed in this paper, a macroblock with a size of 16 * 16 pixels is used for motion compensation. Each macro block in the current frame adopts the minimum SAD (Sum of Absolute Differences) criterion, and searches in the previous frame to find the macro block corresponding to the current macro block with the smallest SAD value. This macro block corresponds to the previous frame. The macro block with the most matching current macro block is called the reference macro block, and its definition is shown in formula (21).

$$\mathrm{SAD} = \sqrt{\left|\varpi_{ij} - \vartheta_{ij}\right|}. \tag{21}$$

ϖ_{ij} represents the gray value of each pixel in the current macro block, and ϑ_{ij} represents the gray value of each pixel in the reference macro block. Using the motion compensation technology based on the minimum SAD, in the fixed search range of the previous frame, the most matching reference macroblock corresponding to the current macroblock can be found. The current macroblock and the reference macroblock constitute the motion trajectory of the current macroblock in the time domain, and the current macroblock can be filtered in the time domain along the motion trajectory.

To overcome the “tailing” phenomenon of the fast-moving objects easily caused by pure time-domain filtering, motion intensity detection technology is adopted. For objects with different motion intensities, the filter adopts different filtering intensities, which effectively avoids the phenomenon of “tailing” of fast-moving objects. Since the algorithm is processed with a macro block as the smallest unit in the algorithm, a motion intensity detection operator τ is defined to detect the motion intensity of the current macro block on the motion trajectory. The definition of τ is shown in formula (22).

$$\tau = \sum_{i=1}^{n}\sum_{j=1}^{n} \varpi_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} \vartheta_{ij}. \tag{22}$$

By the test of a large number of motion sequences, two empirical values for measuring the motion intensity of the macroblock are determined, namely the high and low thresholds h_1 and h_2 of the motion intensity of the macroblock. By detecting the operator τ and the two high and low thresholds h_1 and h_2 determined by experiments, the motion intensity of the current macroblock on the motion trajectory can be determined [15].

The motion intensity of each macro block is defined as three cases. If the value of the detection operator τ of the current macro block is less than the low threshold value h_1, it indicates that the current macro block is a steady and slow motion on the motion trajectory with low motion strength. In this case, the filter intensity of the filter can be set higher, which can effectively remove noise, and at the same time, because of the low motion intensity of the current block, it will not cause the phenomenon of “tailing.” When the value of the detection operator τ is higher than the high threshold value h_2, it indicates that the current macroblock is violently moving on the motion trajectory and has a strong motion intensity. At this time, the filter strength should be adjusted to a lower level so that the filtered macroblock keeps the information of the current macroblock as much as possible to avoid the “tailing” phenomenon. When the value of the detection operator τ is between h_1 and h_2, it indicates that the motion intensity of the current macroblock is in an intermediate state, and the filtering intensity of the filter will also be adjusted to an intermediate level. The filter strength of the filter is adjusted by the corresponding weight λ. The definition of λ is as shown in formula (23).

$$\begin{cases} \lambda = 0.35, & \tau \leq h_1, \\ \lambda = 0.6, & h_1 < \tau < h_2, \\ \lambda = 0.8, & h_1 \leq \tau. \end{cases} \tag{23}$$

According to the motion intensity of the macroblock, the rapid compensation of the motion between the multimedia video frames can be realized by adjusting the weight of the filter intensity.

4. Simulation Experiment

To verify the effectiveness of the fast interframe motion compensation algorithm of the multimedia video based on the Manhattan distance proposed in this paper, the compensation algorithm based on depth learning and the compensation algorithm based on depth convolution neural network are used as comparison methods, and the application effects of different methods are judged according to the experimental results.

4.1. Experimental Platform and Parameter Settings. The working platform parameters of this experiment are as follows: the processor is Intel Pentium(R) Dual-core CPU E6500 2.93 GHz, the memory is 2 GB, and the operating system is Windows XP Professional. In the experiment, the JVT-released H.264 standard JM12.4 version of the official codec software was compiled and implemented on the Visual C++ software platform, and the JCT-VC-released HEVC standard HM9.0 version test model was compiled and implemented on the Visual Studio 2008 software platform. In the experiment, four official test sequences with different characteristics were used to complete the comparative experiment, as shown in Table 1.

To comprehensively compare the performance differences between various methods, four groups of the test video sequences with different motion amplitudes, motion directions, and numbers, as well as the size of the moving objects are selected as the experimental data objects. These four groups of video sequences have different objects and different motion modes. Sequence 1 is mainly vertical movement, and the moving objects are small, and also, the range of motion is small. Sequence 2 mainly shows horizontal movement, however, the moving objects are larger and the movement range is small, especially the movement of large cargo ships. Sequence 3 mainly reflects the movement of the coast guard motorboats and yachts in the horizontal direction, and the motorboats have a larger range of motion, while the yacht movement range is small. Sequence 4 is mainly reflected in the horizontal direction of the car movement, and the movement range is relatively large. At the same time, because of the lack of vertical motion changes in the standard video sequence, this article took two pictures. The characteristics of these two video sequences are that there are different amplitudes of motion in the vertical direction in the image. Self-portrait 2 is larger than self-portrait 1.

For the above four test sequences, the compensation effects of different methods are counted and compared. The experimental results are shown below.

4.2. Analysis of Experimental Results

4.2.1. Peak Signal-to-Noise Ratio PSNR. PSNR is an objective evaluation method that can reflect the actual visual effect of the video in general. The calculation process is relatively simple, and it is widely used in the fields of video coding and image processing. The calculation formula of PSNR is as follows:

$$\mathrm{PSNR} = 101\mathrm{lg}\frac{\left(2^n - 1\right)^2}{\mathrm{MSE}}. \tag{24}$$

MSE (Mean Square Error) is the mean square error between the original video image and the processed video image. The unit of PSNR is expressed in decibels (dB). Under normal circumstances, the greater the PSNR value, the closer the quality of the processed video image will be to the original video image. In some special cases, the PSNR value is too large and the actual effect of the video image is poor. The peak signal-to-noise ratio comparison results of different methods are shown in Table 2.

By analyzing the data in Table 2, it can be seen that the peak signal-to-noise ratio of the video image is higher than that of the depth learning algorithm and depth convolution neural network algorithm after using this method to compensate the application between video frames in different test sequences. Although the peak signal-to-noise ratio of the depth convolution neural network algorithm is better than that of the depth learning algorithm, it still has a certain distance from this algorithm. Therefore, after using this algorithm to compensate for the motion between the video frames, the quality of the video image has been effectively improved, which shows that this method has a better application effect.

4.2.2. Visual Effect Evaluation. Firstly, evaluate the results of the interframe motion compensation of the video image by the algorithm in this paper, the deep learning algorithm, and the deep convolutional neural network algorithm from the perspective of visual effects. Figure 6 shows the effect of the three algorithms for motion compensation on the representative images of test sequence 1 to generate interpolated frames.

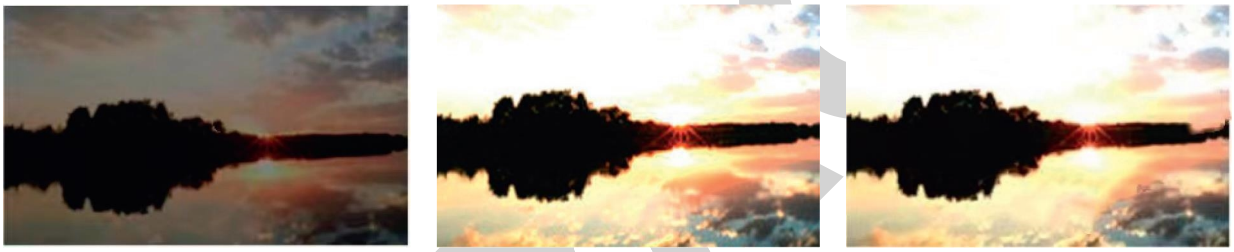
According to Figure 6, it can be seen that the three algorithms perform motion interpolation on the input video image well. Note that the deep convolutional neural network algorithm has a slight loss in image details. Although the deep learning algorithm maintains more video image details than the deep convolutional neural network algorithm, there are some errors in interpolation. Compared with the two traditional algorithms, the visual effect of the algorithm in this paper is better. There is no loss of details, and the clarity is higher. From the visual effect evaluation results, the

Table 1: Characteristic parameters of comparative experimental test sequence.

Test sequence	Resolution	Total number of frames	Video characteristics		
			Details	Color	Movement
1	352 × 288	300	Average	Average	Small
2	352 × 288	500	Average	Average	Small
3	416 × 240	400	Less	Average	More
4	416 × 240	500	Less	Average	Average

Table 2: Peak signal-to-noise ratio.

Test sequence	Peak signal-to-noise ratio/dB		
	The algorithm of this article	Deep learning algorithm	Deep convolutional neural network algorithm
1	12.08	9.56	10.21
2	25.47	21.37	23.67
3	51.39	46.25	47.52
4	9.70	6.94	7.26



(a) (b) (c)

Figure 6: Visual effect evaluation. (a) Deep learning algorithm. (b) Deep convolutional neural network algorithm. (c) The algorithm of this paper.

algorithm in this paper can perform correct motion interpolation on the video image sequence, which shows that the algorithm in this paper has good generalization ability.

4.2.3. Execution Time. The execution time of video motion compensation is used as an experimental indicator, and different methods are compared. The results are shown in Figure 7.

Analyzing Figure 7 shows that when using the algorithm in this paper to perform motion compensation on multimedia video images, the execution time is always less than 0.5 s, and the minimum is only 0.25 s, while deep learning algorithms and deep convolutional neural network algorithms are used to perform motion compensation on multimedia video images. The execution time is much higher than the calculation time of the algorithm proposed in this paper. It can be seen that the execution time of the algorithm in this paper is shorter, and the motion compensation of the video image can be realized at a faster speed.

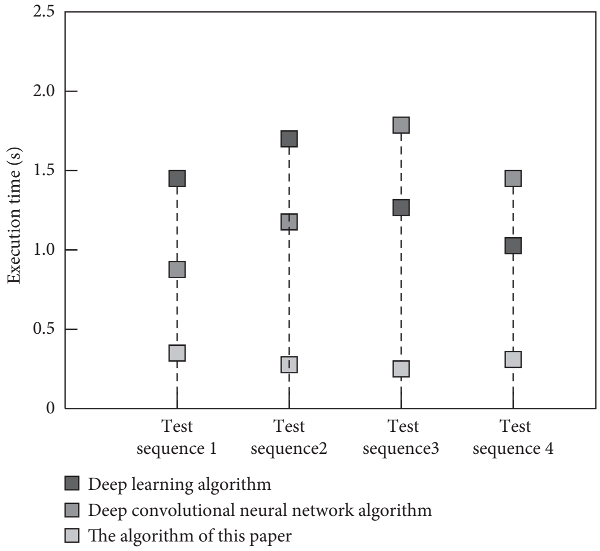


Figure 7: Comparison results of execution time.

4.2.4. Bit Rate. The higher the bit rate, the better the video image quality and the smaller the distortion. Taking test sequence 3 and test sequence 4 as examples, the bit rate is used as an experimental indicator to compare the video compensation effects of different methods. The results are shown in Figure 8.

According to Figure 8, it can be seen that in the test of test sequence 3 and test sequence 4, the code rate of this algorithm is higher than that of the deep learning algorithm

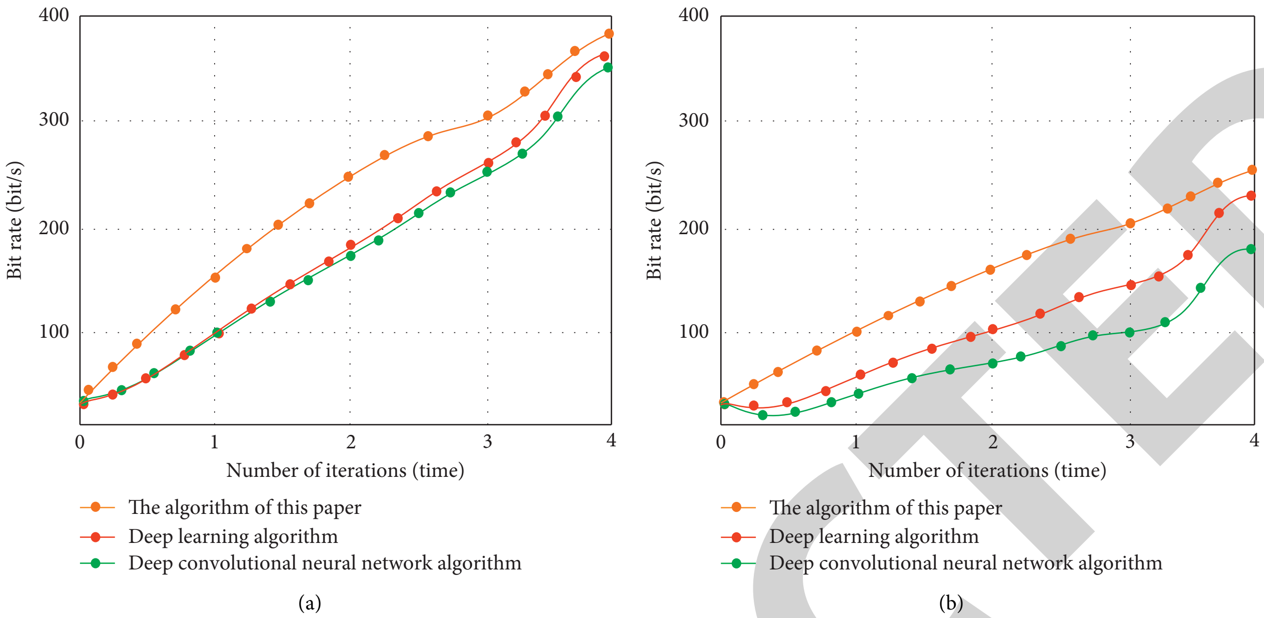


Figure 8: Code rate comparison result. (a) Test sequence 3. (b) Test sequence 4.

and deep convolutional neural network algorithm. It shows that the better the video image quality of the algorithm in this paper, the smaller the distortion, which further verifies the application value of the algorithm in this paper.

5. Conclusion

To solve the problems of low peak signal-to-noise ratio, poor visual effect, and low bit rate in traditional methods, this paper proposed a fast compensation algorithm for motion between multimedia video frames based on Manhattan distance. The purpose of denoising and extracting significant information from the video image was achieved by the preprocessing of the video image. To this end, the block-based motion idea was the division of each frame of the video sequence into nonoverlapping macroblocks. Moreover, the best position of the block corresponding to the current frame was found. Furthermore, the current frame of the multimedia video was obtained with the Manhattan distance calculation formula. Then, the compensation of the relative Manhattan distance between the backgrounds was pushed forward for the motion between the multimedia video frames. Finally, the experimental results were analyzed, which verified that the proposed algorithm has a high peak signal-to-noise ratio, a higher bit rate, and a better video visual effect.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The paper did not receive any financial support.

References

[1] A. J. Hussain and Z. Ahmed, “A survey on video compression fast block matching algorithms,” *Neurocomputing*, vol. 335, no. 28, pp. 215–237, 2019.

[2] B. Veerasamy and S. Annadurai, “Video compression using hybrid hexagon search and teaching–learning-based optimization technique for 3D reconstruction,” *Multimedia Systems*, vol. 27, no. 12, pp. 1–15, 2021.

[3] M. N. Sadat, R. Dai, L. Kong, and J. Zhu, “QoE-aware multi-source video streaming in content centric networks,” *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2321–2330, 2019.

[4] P. Maniotis, E. Bourtsoulatze, and N. Thomos, “Tile-based joint caching and delivery of 360° videos in heterogeneous networks,” *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2382–2395, 2020.

[5] S. L. Yang, J. Sun, Z. Yan, and X. L. Li, “Research on image interframe compensation based on deep convolutional neural network,” *Computer Simulation*, vol. 37, no. 1, pp. 452–455, 2020.

[6] A. Rezapour, A. Ortega, and M. Sahimi, “Upscaling of geological models of oil reservoirs with unstructured grids using lifting-based graph wavelet transforms,” *Transport in Porous Media*, vol. 127, no. 3, pp. 661–684, 2019.

[7] J. Montalvo, “Noise variance estimation through joint analysis of intrinsic dimension and differential entropy,” *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1330–1333, 2019.

[8] H. M. Hossein-Abad, M. Shabanian, and I. A. Kazerouni, “Fuzzy c-means clustering method with the fuzzy distance definition applied on symmetric triangular fuzzy numbers,” *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 3, pp. 1–15, 2020.

[9] Y. Zhu, G. Zhai, X. Min, and J. Zhou, “The prediction of saliency map for head and eye movements in 360 degree

images," *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2331–2344, 2020.

[10] N. Boonthep, K. Chamnongthai, and P. Phensadsaeng, "Correction to: H.264 video coding-based motion estimation architecture for video broadcasting from a Studio," *Wireless Personal Communications*, vol. 117, no. 2, Article ID 1713, 2021.

[11] S. Chaudhary, A. Dudhane, P. W. Patil, S. Murala, and S. Talbar, "Motion estimation in hazy videos," *Pattern Recognition Letters*, vol. 150, no. 3, pp. 130–138, 2021.

[12] D. Konstantinidis, T. Stathaki, and V. Argyriou, "Phase Amplified correlation for improved sub-pixel motion estimation," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 3089–3101, 2019.

[13] V. V. Chetlur, H. S. Dhillon, and C. P. Dettmann, "Shortest path distance in manhattan Poisson line cox process," *Journal of Statistical Physics*, vol. 181, no. 6, pp. 1–22, 2020.

[14] S. S. Khan, Q. Ran, M. Khan, and M. Zhang, "Hyperspectral image classification using nearest regularized subspace with manhattan distance," *Journal of Applied Remote Sensing*, vol. 14, no. 3, p. 1, 2019.

[15] M. F. Mo, "Simulation of dynamic compensation method for 3D video frame jump frame missing," *Computer Simulation*, vol. 38, no. 3, pp. 135–138, Article ID 179, 2021.