

Research Article

Embedded Image and Video Coding Algorithm Based on Adaptive Filtering Equation

Zhe Fu 

Henan Institute of Economics and Trade, Zhengzhou, Henan 450000, China

Correspondence should be addressed to Zhe Fu; fuzhe@henetc.edu.cn

Received 27 July 2021; Accepted 21 August 2021; Published 9 September 2021

Academic Editor: Miaochoao Chen

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

Based on the improved adaptive filtering method, this paper conducts in-depth discussion and research on embedded graphics and video coding and chooses to improve the adaptive filtering algorithm from three aspects: starting point prediction, search template, and window partitioning. The algorithm is imported into the encoder for video capture and encoding. By capturing videos of different formats, resolutions, and times, the memory size of the video files collected before and after the algorithm optimization is compared, and the optimized algorithm occupies the memory space of the video file in the actual system. The conclusion of less and higher coding rates. The collected video information is stored on a personal computer equipped with a freeness, and external electronic devices only need to download and install the browser, and the collected video information can be accessed in the local area network through the protocol. The improved coding algorithm has higher coding efficiency and can reduce the storage space occupied by the video.

1. Introduction

Image information occupies the main part of the information that people obtain from the outside world through vision. Therefore, the importance of intuitive information provided by images is self-evident. Video images are continuous static image sequences that can make objective things more vivid and vivid. The image description is a more intuitive and specific form of information expression [1]. With the rapid development of computer science and the improvement of people's security awareness, video image acquisition equipment can be seen everywhere, and the video surveillance system composed of video acquisition equipment and computers has attracted increased attention due to its intuitive and convenient features. The current video surveillance system can not only monitor in real-time but also detect, track, and identify moving objects [2]. The prerequisite for realizing these complex algorithms is the edge detection of the collected images. Therefore, the accurate extraction of edge information provides a certain basis for the implementation of subsequent image segmentation and recognition algorithms [3]. At present, more than

90% of Internet traffic is generated by video services, and video data in the mobile Internet also accounts for 50%. Video has a profound impact on our work and life. The video surveillance system composed of video capture equipment and computers is more ubiquitous due to its intuitive and convenient features. In real life, the video has a lot of information redundancy. If it is not effectively compressed, it is difficult to effectively store and disseminate within the current limited bandwidth and storage space [4]. To meet people's demand for high-definition video under current circumstances, the use of video compression technology to reduce the storage capacity and transmission bandwidth of video information has become an urgent and realistic research topic [5].

Botella and Garcia analyzed the algorithms and found that the motion type of the matching block and the search mode has a great correlation [6]. Through the test, the motion type and the corresponding search mode are matched and counted [7]. Next, a new hybrid search algorithm was proposed; Wang and Zhao proposed an improved algorithm based on the difference in motion direction [8]; Shu et al. borrowed from the research results of the

predecessors in the algorithm; a multilayer hierarchical fast-adapting whole-pixel motion estimation algorithm was proposed [9]; Tew et al. aimed at the slow search speed of the algorithm, using the center bias characteristics of the predicted motion vector, and compared the algorithm [10]. Some improvements were made, and an improved motion estimation algorithm was proposed; Dutta and Gupta proposed its improvement methods for the algorithm in four aspects: starting point prediction, search template, search window, and early termination [11]; Coutinho et al. focused on the motion estimation part of the standard, using the motion vector prediction characteristics to obtain the optimal starting point, and setting two thresholds of different sizes to achieve a spiral fast search method [12]. Granado et al. designed a real-time edge detection system based on Sobel operator on the FPGA platform [13]; Hwang et al. increased the detection direction of Sobel operator [14], analyzed and compared the optimization methods of multiple operators, and improved to improve the operating speed of the system; Khursheed and others use the idea of distributed method processing to accelerate the Canny algorithm [15]. Compared with ordinary algorithms, the edge extraction effect is significant and the system consumes less time. For the second derivative operator, De and others analyzed and improved the LOG algorithm [16] and verified it on Altera's DE2 platform. The above edge detection algorithms all use a fixed threshold in the implementation process, that is, artificially set a threshold outside the system according to experience, which is not adaptive, and the algorithm's versatility is poor [17]. For this reason, the threshold selection method of the maximum value of the second derivative of the gradient histogram, although the effect is significant, the algorithm structure is complex and the implementation is difficult [18]. Similarly, for the Canny algorithm, Khan et al. used the Maximum Between-Class Variance Method (Otsu) to find the adaptive threshold [19]. As it involves finding the variance of image pixels, it will consume a lot of logic resources and seriously affect the processing speed of the system. Therefore, seeking a suitable adaptive threshold method that is convenient for hardware implementation has an important impact on the processing speed of the entire edge detection algorithm.

Motion estimation is the core technology of video coding technology, and its performance is directly related to video coding efficiency and image effect. Therefore, selecting a motion estimation algorithm with superior performance has become the focus of scholars' research. There are two common motion representations, one is a pixel-based representation, and the other is block-based motion representation. Because the calculation based on the pixel representation is complicated and the accuracy is not high, the block-based motion representation is usually used. The motion estimation algorithm based on block matching representation has attracted the attention of researchers and has been adopted by various coding standards due to its simple algorithm, excellent effect, and easy implementation. Based on the analysis of the coding algorithm in the coding standard, this paper focuses on the two parts of rate-distortion optimization and motion estimation. By consult-

ing a large amount of data, some ideas for the selection of distortion modes and optimization of adaptive filtering algorithms were put forward and tested in the test model. To test the encoding performance of the improved encoding algorithm on the video system, the algorithm transplantation and video implementation were carried out on the development board. The results show that the improved new algorithm has better encoding performance in the actual system.

2. Design of Adaptive Filtering Embedded Graphic Video Coding

2.1. Improved Adaptive Filter Coding Algorithm. Traditional adaptive filtering algorithms mainly rely on the FIR horizontal structure, and the designed algorithms are mostly linear functions between input and output. At present, more than 90% of Internet traffic is generated by video services, and video data in the mobile Internet also accounts for 50%. This type of filtering algorithm has low computational complexity and has good performance in linear system identification, linear channel regression, active noise suppression, or elimination. However, when linear filtering algorithms deal with nonlinear problems such as nonlinear signal regression, nonlinear system identification, and time series prediction, they will show performance degradation or failure to operate normally [20]. To deal with these nonlinear problems, researchers have developed many related processing methods, such as the Volterra function sequence, time-delay multilayer perceptron, radial basis function network, and recurrent neural network. Although these methods can show better nonlinear processing capabilities, the inherent nonconvex characteristics, slow convergence, and huge computational overhead hinder the performance of these methods in real-time applications [21]. Relying on the kernel method and the theory of adaptive filtering, the kernel adaptive filtering algorithm (KAF) has great development and application in the field of adaptive filtering. Since KAF is a kind of online-nonlinear adaptive filtering algorithm developed in reproducing kernel Hilbert space (RKHS), they can effectively deal with the nonlinear pattern relationship between input and output data, as shown in Figure 1.

The Mercer kernel function needs to satisfy three conditions: continuity, symmetry, and positive definiteness [22].

$$M^H Kc = \lim_{N \rightarrow \infty} \sum_{i=1, j=1}^N c_i^* c_j K_{i,j}, \quad (1)$$

In practical applications, the common kernel functions mainly include

$$K(x, y) = \begin{cases} \langle x, y \rangle, \\ (\langle x, y \rangle + c)^p. \end{cases} \quad (2)$$

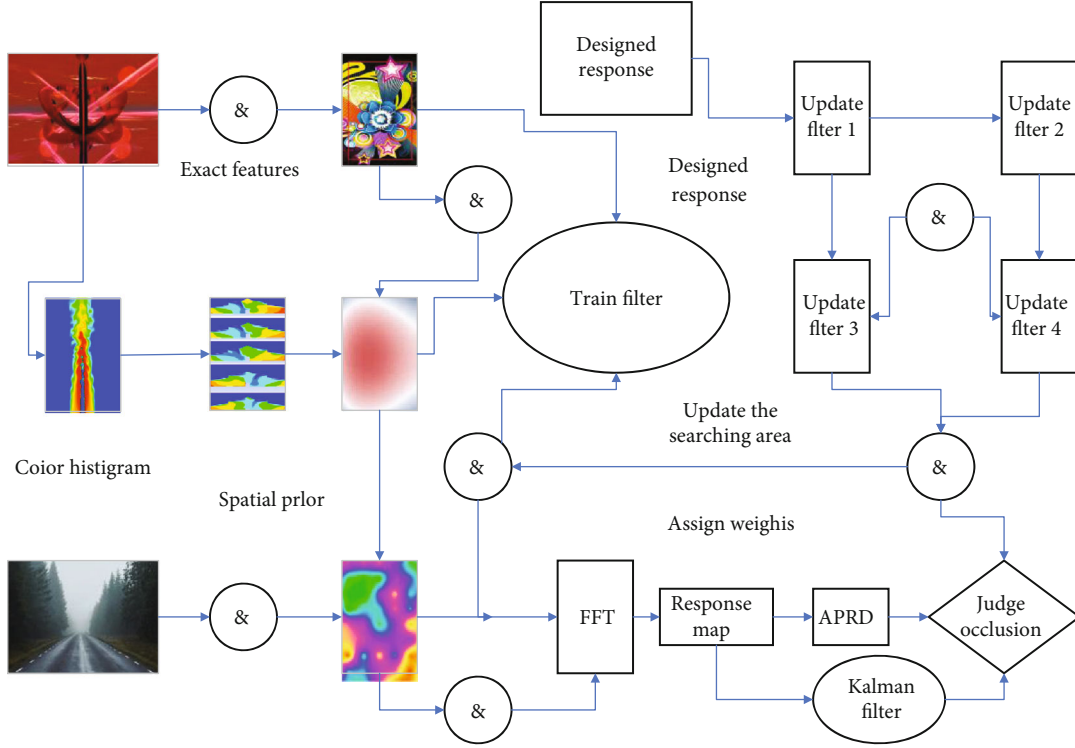


FIGURE 1: The improved adaptive filtering algorithm.

Triangular kernel function:

$$K(x, y) = \begin{cases} (b - \langle x, y \rangle)_2, \\ (\langle x, y \rangle + c)^p. \end{cases} \quad (3)$$

Exponential kernel function:

$$K(x, y) = \begin{cases} \exp(b - \langle x, y \rangle)_2, \\ \exp(\langle x, y \rangle + c)^p. \end{cases} \quad (4)$$

The feature space induced by the Gaussian kernel function has infinite dimensions so that most adaptive filtering algorithms based on the kernel method use the Gaussian kernel function.

Let M denote the function space formed by the kernel function K , and take the elements x and y in M , then

$$K(x, y) = \begin{cases} \lim_{M \rightarrow \infty} \sum_{i=1}^M a_i \exp(b - \langle x, y \rangle)_2, \\ \lim_{M \rightarrow \infty} \sum_{i=1}^M b_i \exp(\langle x, y \rangle + c)^p. \end{cases} \quad (5)$$

If i and j further satisfy the following relationship:

$$\langle h(\cdot), g(\cdot) \rangle = \lim_{M \rightarrow \infty} \sum_{i=1}^M b_i \lim_{N \rightarrow \infty} \sum_{j=1}^N c_i^* c_j K_{i,j}. \quad (6)$$

Formula (6) is called bilinear mapping, which is an inner product operation. Equation (7) can be expressed as

$$\langle h(\cdot), g(\cdot) \rangle = \lim_{M \rightarrow \infty} \sum_{i=1}^M b_i \lim_{N \rightarrow \infty} \sum_{j=1}^N (c_i^* c_j K_{i,j} + x_i). \quad (7)$$

According to Mercer's theorem, a certain nonlinear function can express any Mercer kernel function:

$$K(x, y) = \langle \phi(\cdot), g(\cdot) \rangle, \quad (8)$$

$$K(x, y) = \lim_{M \rightarrow \infty} \sum_{i=1}^M b_i \lim_{N \rightarrow \infty} \sum_{j=1}^N (c_i^* c_j K_{i,j} + x_i) \langle \phi(\cdot), g(\cdot) \rangle.$$

This formula shows that the nonlinear operation in the original space can be expanded to the linear operation in the high-dimensional space. Besides, the vector dimension in the high-dimensional space is much higher than the original space, and more regression factors can be used to solve the nonlinear problem in the original space. Therefore, it is very important to choose a kernel function with increased size in KAF applications.

For the KAF algorithm, the usual goal is based on a set of training samples to reconstruct. You can get

$$\omega(0) = 0. \quad (9)$$

Among them, $\omega(n)$ means the prediction error at n

moment, n means the learning step length, and e to ensure the convexity of MPE. To further expand the third term of formula (9), there can be

$$\omega(n) = n \lim_{M \rightarrow \infty} \sum_{i=1}^M b_i |e(i)|^2. \quad (10)$$

In practical applications, since the nonlinear transformation corresponding to the kernel function is usually unknown, equation (10) is only used as a theoretical expression. However, when a certain new data appears, for example, at this time, the estimated output can be obtained as

$$F_n(x) = \langle \omega(\cdot), \lambda(\cdot) \rangle. \quad (11)$$

2.2. Design of Embedded Image and Video Coding System. As shown in Figure 2, JetsonTX2 is equipped with a powerful processor, memory, flash memory, and other aspects and rich peripheral interfaces, ensuring its powerful data processing and computing capabilities, so it can be used in fields such as drones and human intelligence. According to the application requirements of this thesis, JetsonTX2 will be introduced from four aspects: video input, parallel acceleration, video processing, and network communication.

The module is divided into 6 parts: PCI interface design and register configuration module, video image acquisition module, improved Canny edge detection algorithm, control module, display module control, and internal phase-locked loop generation module. First, a phase-locked loop (PLL) module is used to generate a 24MHz camera drive clock. The camera is driven through the design of the PC interface, the initialization of the camera's internal registers, and the design of the video image acquisition module so that the camera output data stream is 565 bits wide. After the 16-bit image data is processed by the threshold adaptive edge detection algorithm, the data is buffered through synchronous dynamic random-access memory (SDRAM), and finally, the real-time edge detection image is obtained through the display. Among them, based on the improved Canny edge detection algorithm module, it includes gray-scale transformation, adaptive median filtering, calculation of gradient amplitude and direction, nonmaximum suppression processing, edge connection, adaptive threshold generation, and other modules. Because the calculation based on the pixel representation is complicated and the accuracy is not high, the block-based motion representation is usually used. The motion estimation algorithm based on block matching representation has attracted the attention of researchers and has been adopted by various coding standards due to its simple algorithm, excellent effect, and easy implementation. This chapter only analyzes and designs the collection of real-time video images, that is, the 8-bit data collected by the camera is spliced into the format of RGB 565 by the image acquisition module and directly buffered by SDRAM and finally displayed in real-time on the Video Graphics Array (VGA) monitor through the VGA display control.

After analyzing the read and write timing of PCI, the PC interface is now used to configure the registers in the SDRAM [23]. SDRAM has a total of 172 registers, and each register has a corresponding address and a configuration value of the register. The configuration value of the register is composed of 8-bit data, and the corresponding parameters can be modified to achieve different functions. If you want to output the expected video format, you must configure these registers according to the required functions. However, the number of these 172 registers is large, and all the configuration is more troublesome. The design of this article only uses the more important 70 registers. Including two read-only registers, video image stream, sampling image quality, output format, and other important registers. Since the required video output format is RGB 565, the image display is VGA format, and the resolution is 1920 * 1080; it is necessary to modify the value of the default register to meet the design requirements. Use look-up table technology to configure these required registers. Splice the address value of these 70 registers with the configuration value of the register, and use the signal to find the number of register configurations currently required by the PCI interface module. It represents the address of the register and the configuration of the register.

The kernelized correlation filter tracking algorithm is a target tracking algorithm proposed based on the correlation filter theory. The core of the current mainstream tracking algorithm is to design a classifier with a strong discriminative ability to distinguish the target and the background interference around the target. In the process of changes in the appearance of the target and the environment, the traditional classifier uses training samples generated by panning and scaling the initial target to train the classifier. The strategy for generating samples is called a sparse sampling strategy, and there are many redundant samples. In correlation filtering tracking, the sample generation strategy is based on dense sampling. Many training samples are generated through the cyclic shift operation of the samples. The samples generated by the cyclic shift are expressed as a cyclic shift matrix. In the theory of correlation filtering, the cyclic shift matrix has the property that can be diagonalized by the Fourier transformation matrix, through a series of theoretical derivation and simplification; the correlation filter tracking algorithm is mainly calculated as fast Fourier calculation, which greatly reduces the calculation amount of the tracking algorithm and the amount of storage required by the algorithm. Besides, KCF has also greatly improved the tracking accuracy by introducing nuclear techniques and multichannel HOG features.

2.3. Evaluation Index Design. As a very important part of video coding, the rate-distortion algorithm of video coding has become the research object of many researchers. They have put forward many excellent ideas for the improvement of this algorithm. The goal is to obtain the best coding mode. In this mode, the bit rate R and coding distortion D minimize the coding cost J . Under the standard, rate-distortion calculation is performed on the cost of each mode used by the current coding block, and then, each mode is compared

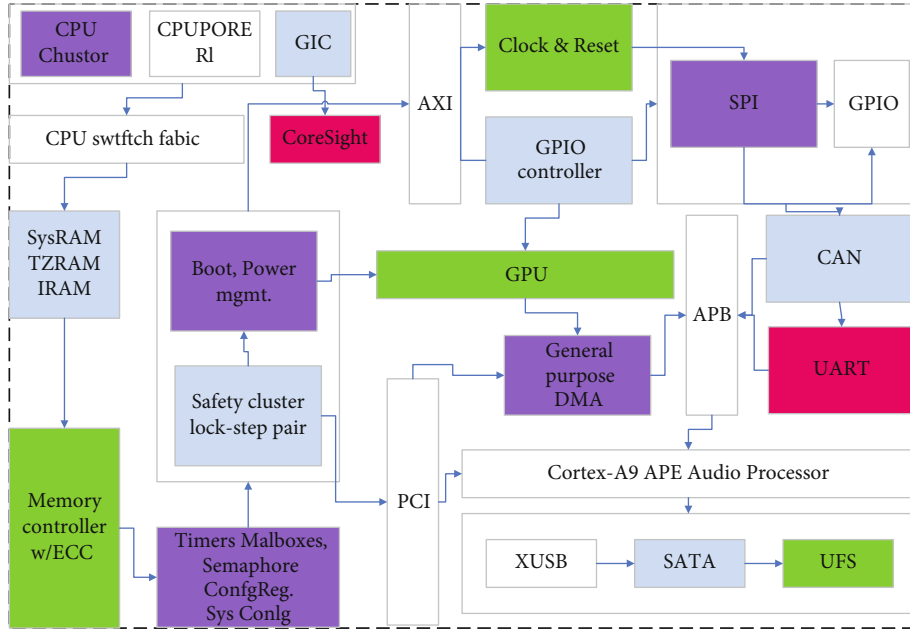


FIGURE 2: The internal structure of JetsonTX2.

and analyzed, and the type of coding mode that consumes the least cost is selected, and the selected coding mode is defined. It can be seen from the cost function that its cost value is jointly determined by three factors: motion search, reference frame selection, and mode decision-making, while the standard just uses traversal calculation to perform the rate. 1Distortion optimization does not fully consider the influence of other factors.

In response to this problem, this article proposes a little improvement idea on how to confirm the selection algorithm of intermacroblock coding mode in the original rate-distortion optimization. There are many ways to divide macroblocks between frames. According to the degree of motion, we can make different divisions. The large division method is suitable for absolute static or relatively static small motion, and the small division method is suitable for large changes in position and details. This scheme comprehensively considers the constraints of multiple factors and finally adopts an $8 * 8$ division of the block to be coded, divides the block to be coded into $48 * 8$ modes, and performs corresponding motion vector direction analysis on the submodules obtained by $8 * 8$ mode division. From these results, the coding modes are currently unusable in the macroblocks to be coded, so those unnecessary calculations can be reduced. Then, the set of coding modes that may be used in the coding block is created together to create a coding mode set, and the rate-distortion optimization calculation is performed on it to obtain the best mode required. The test shows that the improved rate-distortion algorithm has a certain degree of coding efficiency, as shown in Figure 3.

The full search method (FS) is currently the most accurate search algorithm. It searches for all search points to get the best matching point. Due to the high computational complexity, it is not suitable for transmission in real-time video and can only be used as a comparison standard for

other algorithms. The three-step method (TSS) is simplified based on the full search method. In the fastest case, only 25 search points are needed to obtain the best matching point. Although the computational complexity is reduced a lot, at the same time, the matching accuracy is also reduced. When nonlinear problems such as nonlinear signal regression, nonlinear system identification, and time series forecasting are processed, performance degradation or failure of normal operation will appear. To deal with these nonlinear problems, researchers have developed many related processing methods, such as Volterra function sequence, time-delay multilayer perceptron, radial basis function network, and recurrent neural network. Although these methods can show good nonlinear processing capabilities, the inherent nonconvex characteristics, slow convergence, and huge computational overhead hinder the performance of these methods in real-time applications. The three-step method is only suitable for frame images with large motion amplitude. For frame images with small motion amplitude, this algorithm is easy to fall into the state of the local optimal solution, resulting in a larger match. The new three-step method (NTSS) is a supplement to the shortcomings of the original three-step method (TSS). The algorithm uses the center offset feature to enhance the number of matching calculations for the center area position, and the search performance is improved. Smaller video sequences have good performance. At the same time, the algorithm innovatively proposes a method of quoting thresholds, which provides a new idea for later hybrid search algorithms. The hexagonal search method (FSS) and the diamond search method (DS), as classic block matching motion estimation algorithms, use the same idea and use two different search templates to avoid the defects of local optimization. The Hexagons algorithm can be regarded as an improvement based on the hexagonal search method (FSS) and the diamond search method (DS).

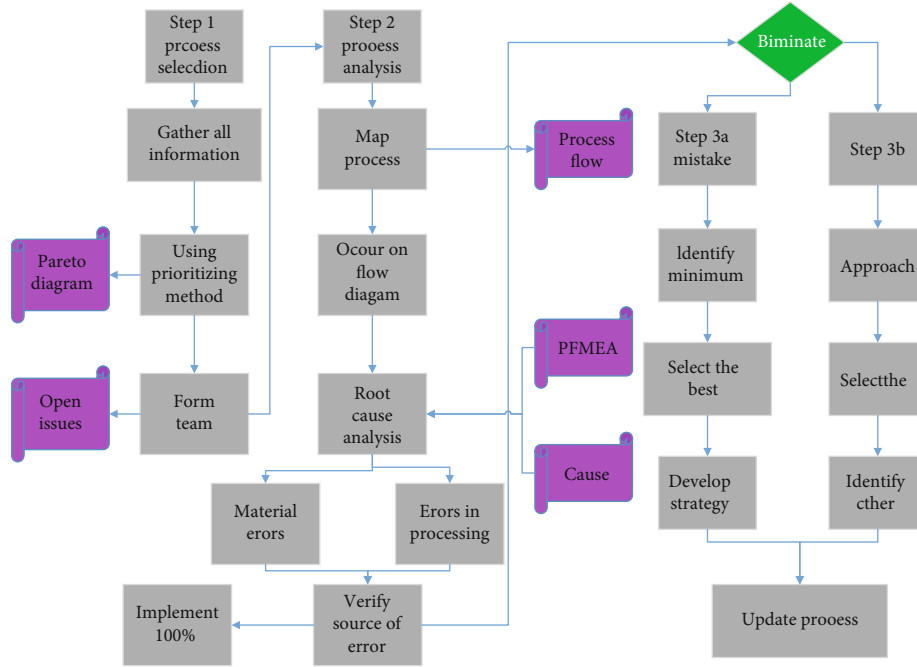


FIGURE 3: Video coding rate distortion evaluation.

TABLE 1: The average probability statistics of the accurate prediction starting point corresponding to the five different prediction methods.

Number of reference frames	Median forecast	Upper prediction	Origin prediction	Previous frame prediction	Prerference frame prediction
1	75.6	71.6	76.8	71.1	77.6
2	79	71	72.6	74.5	73.6
3	72.8	72.6	72.4	75.2	78.9
5	76.9	76.1	79.6	73.1	71.5

The UMHexagonS algorithm uses a hybrid search mode combining multiple templates, combined with an early termination strategy, and can precise prediction of search points be currently recognized as the most ideal motion estimation algorithm under the standard. An important work point in this paper is also to propose its ideas for the improvement of the Hexagons algorithm, as shown in Table 1.

The use of different numbers of reference frames will affect the probability of successful prediction. Among them, the probability of accurately predicting the starting point is the highest when the median and upper-level predictions select different numbers of reference frames. The prediction of the previous frame and the previous reference frame in the time domain will occupy a large amount of storage space. Most of the adaptive filtering algorithms based on kernel methods use Gaussian kernel functions. The prediction method in the spatial domain can meet the requirements of the starting point of successful prediction. On this basis, this paper finally decided to use only median prediction, upper layer prediction, and origin prediction in the spatial domain, and after median prediction and upper layer prediction, the function was added for early termination judgment.

3. Results and Analysis

3.1. Analysis of Adaptive Filtering Target Tracking Results. In the field of target tracking, the tracking result evaluation index usually adopts the precision measurement (precision plot), which refers to the Euclidean distance between the center point of the algorithm predicted position in the target tracking and the actual center position in the ground truth standard, and the unit is the pixel. To test the performance of the tracking algorithm before and after the improvement, this paper uses the algorithm before and after the improvement to simulate all the above-mentioned test video sequences and save the results for comparison. Figure 4 uses the accuracy measurement curve and success rate curve of the improved algorithm. This curve compares the tracking accuracy of the original kernel correlation filter KCF, KCF combined with APEC indicators, and the tracking algorithm after adding the SVM online classifier on the test data set and tracks the success rate. According to the graph, the introduction of APEC index judgment based on the original KCF significantly improves the tracking accuracy of the algorithm; after the introduction of APEC+SVM based on KCF tracking, the tracking accuracy has been improved to a certain extent compared with only using APEC indexes.

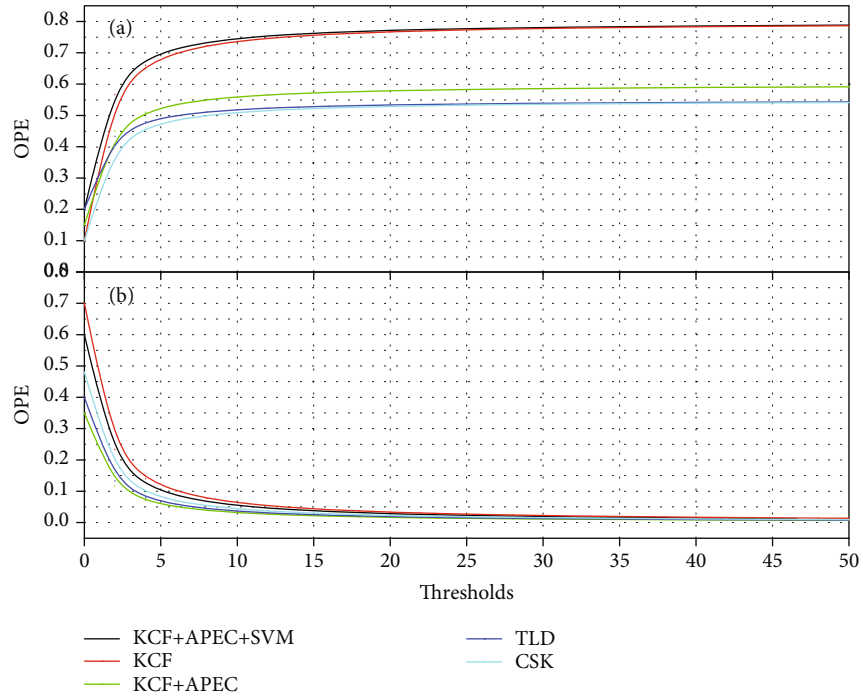


FIGURE 4: Algorithm curve before and after improvement.

Comparing the tracking results of the algorithm after kernelization correlation filter tracking and antiocclusion improvement, this video sequence comes from the data set, and the target will undergo illumination changes and occlusion during operation. As shown in Figure 4, the target is completely occluded in 255~278 frames. The KCF algorithm does not judge the confidence of the tracking result and adopts the strategy of updating the model every frame. As a result, the model learns the occlude due to the wrong update during the complete occlusion period. Even if the target leaves the occlude in frames 278~286, the KCF algorithm is wrong. The value of the default register needs to be modified to meet the design requirements. Use look-up table technology to configure these required registers. The address value of these 70 registers and the configuration value of the register are spliced together, and the number signal of the current required register configuration output by the interface module is used to search, which indicates the address of the register and the data after the splicing in the configuration of the register. As a result, the update leads to subsequent tracking failures; to analyze the performance of the improved algorithm, the algorithm update curve is drawn in real-time in the experiment, as shown in Figure 5; the shape of the graph is a “square wave” shape, where 0 means that the model is not updated and 1 represents the model update. From the figure, the target is occluded during the period of 255~278 frames. Since the improved algorithm judges the confidence of the tracking result, the model update stopped during the target completely occluded, thereby avoiding the model degradation problem.

The tracking results of the original KCF algorithm and the KCF tracking algorithm after the introduction of APEC indicators are compared. The blue box is the original KCF

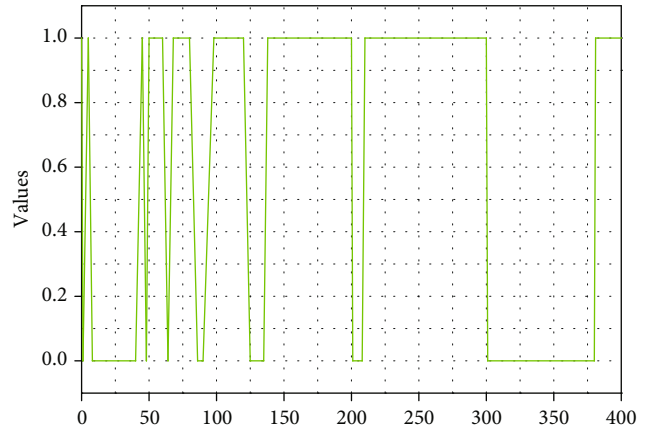


FIGURE 5: The updated curve of the improved algorithm model.

tracking result, and the yellow box is the KCF tracking result after the APEC indicator is introduced. Since APEC index judgment requires statistics of historical values and then compares the current APEC with the historical average APEC, the model is updated in the first 30 frames before APEC is used to determine the confidence of the tracking results, as shown in Figure 6; the target in the first 100 frames is not good. Therefore, the APEC value is relatively large overall; around 133 frames, the target starts to enter the occlusion, and the APEC value drops rapidly currently. As shown in Figure 6, the model stops updating after 100 frames; the original KCF keeps the model updated during the occlusion period. Therefore, the tracking accuracy at the position of 192 frames is significantly worse, and only the part of the target is tracked; both tracking are lost during the 192~235 frames, but the algorithm introduced by APEC

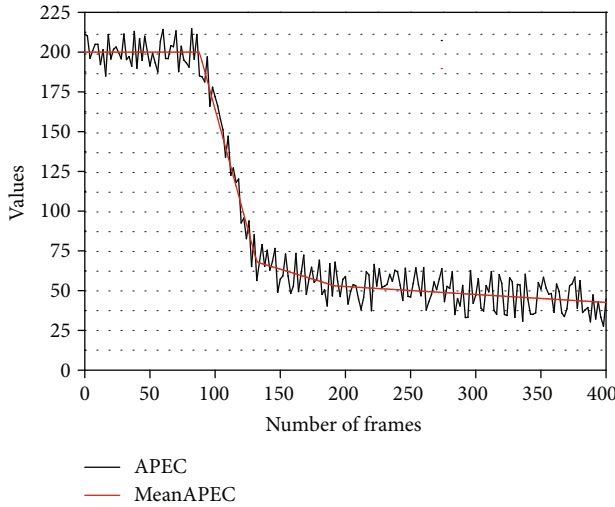


FIGURE 6: APEC, historical average APEC change curve.

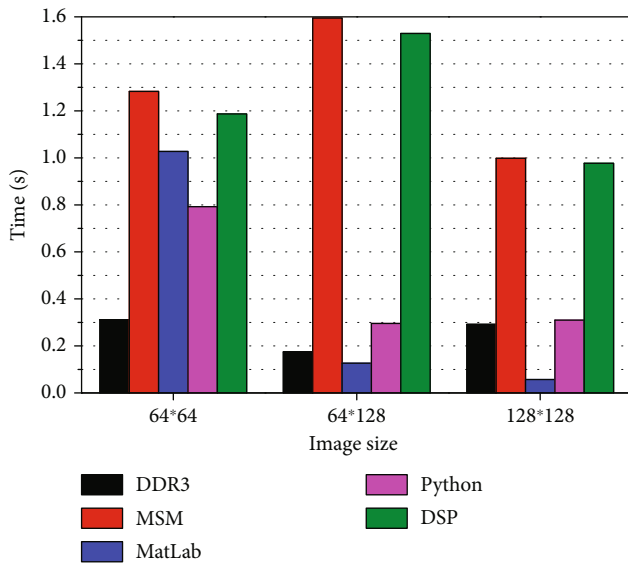


FIGURE 7: Two-dimensional FFT calculation time statistics of images of different sizes and comparison of FFT calculation time of different sizes on PC and DSP.

did not update the model during 377~389 frames; the target moved in the opposite direction. After the introduction of APEC, the KCF still maintained the model before the occlusion. Therefore, the target was retracked during the approach of the target in frame 377, but the original KCF failed to track.

In the core correlation filtering tracking algorithm, the main calculation is FFT, so the calculation speed of FFT has a great influence on the real-time performance of the tracking algorithm. For this experiment, the calculation time consumed by FFT on the DSP is counted. The DSP end clock frequency is 1.0 GHz. In the target tracking, this article mainly uses a 64×128 tracking gate. The 64×128 size image on the DSP two-dimensional FFT calculation time is

0.753 ms, which meets the requirements of the algorithm for real-time calculation. As shown in Figure 7, to compare the calculation performance of FFT on PC and DSP, this paper uses MATLAB and Python NumPy on the PC to calculate the two-dimensional Fourier transform of three different sizes of images. The sample generation strategy is based on dense sampling. Many training samples are generated through the cyclic shift operation of the sample. The sample generated by the cyclic shift is represented as a cyclic shift matrix. In the related filtering theory, the cyclic shift matrix has a Fourier transform matrix. The nature of diagonalization. In the program, the calculation loops 1000 times, and the calculation of the time is averaged. On the PC side, because the computing platform is Intel multicore CPU, the calculation speed is very fast, and the DSP is not as fast as the PC side due to power consumption and other reasons. Although the calculation speed of FFT is also very high, it is beneficial to the real-time target tracking algorithm on the embedded side.

In the kernelized correlation filter tracking algorithm, to improve the accuracy of target tracking, HOG feature extraction is performed on the target area during the calculation process. The HOG feature is a gradient feature and has the advantage of strong robustness. After adding the HOG feature to the target, the accuracy of tracking significantly improved, so the realization of the target tracking algorithm based on HOG features on the DSP side is a great engineering practical value. Since the calculation efficiency of the HOG feature directly affects the time consumed by the target tracking algorithm, this paper experiments to calculate the time consumed by different sizes of images. The experiment uses images of three different sizes, 64×64 , 64×128 , and 128×128 . The gradient direction calculated in VLIB is the unsigned gradient direction, and the maximum setting range of bins is 180. To calculate the best performance of the gradient and gradient direction calculations, the L1P and L1D caches are set to open in the experiment, and all the calculation data are stored in the L2 of the DSP. The access speed of the L2 is compared to the MSM multicore shared memory, DDR3 fast. In this paper, the target tracking gate size is generally 64×128 , and the calculation of the HOG feature gradient direction of the 64×128 image only takes about 1 ms, which meets the real-time requirements of the tracking algorithm. In the kernelized correlation filter tracking algorithm, instead of directly using HOG features, it uses a variant of HOG features—First Home Owner Grant (FHOG) features. FHOG features have 31 dimensions. This article is based on the calculation of HOG features and DSP FHOG feature extraction implemented at the end. In the experiment, the input image is a 128×64 single-channel grayscale image.

3.2. Analysis of Coding Performance Evaluation Results. In the parameter setting of experiment 1, set $K = 10$ to discuss the impact of the asynchronous length on the performance of the algorithm. The related experimental results are shown in Figure 8. Although a smaller step size will reduce the convergence speed of the algorithm, it will improve the filtering accuracy of the algorithm. On the contrary, a larger step size

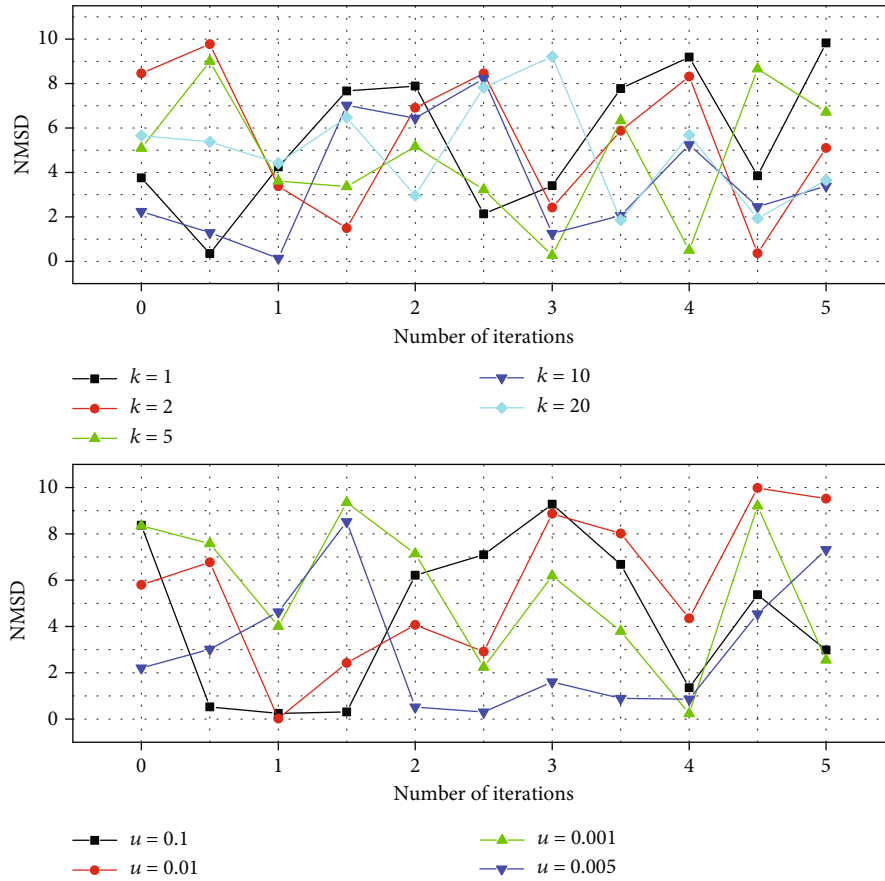


FIGURE 8: Filter coding curve.

speeds up the convergence performance of the algorithm at the expense of filtering accuracy.

The other parameters are the same as the previous settings, to study the influence of different filter lengths on the algorithm. The related experimental results are shown in Figure 8. Increasing L will not significantly improve the filtering accuracy of the algorithm, but it will reduce the convergence speed of the algorithm. At the same time, increasing L will also increase the computational complexity of the algorithm, so the filter length should be selected reasonably.

The signal-to-noise ratio (SNR) is one of the important indicators for evaluating the performance of an algorithm. The signal SNR value improved through the filtering process can be used to quantify the effectiveness of the filter in suppressing motion artifacts. Figure 9 shows the SNR comparison of the suppression results of motion artifacts in ECG signals. This shows that the motion artifact suppression method based on the wavelet adaptive algorithm has better SNR than the traditional adaptive filtering algorithm. This method has better performance in improving the signal's SNR value and suppressing motion artifacts.

Figure 9 shows the spectrum of the ECG signal filtered by various algorithms. It can be seen from the figure that the signal frequency is mainly concentrated in 0~50 HZ. The LMS algorithm and the NLMS algorithm still have some

motion artifact noise after filtering, and the filtering effect is better. The BNLMS algorithm, wavelet LMS algorithm, wavelet NLMS algorithm, and wavelet BNLMS algorithm have obvious effects in suppressing motion artifact noise. Compared with the filtering results of the traditional LMS and NLMS algorithm, the motion artifact noise is significantly reduced, but it still has a small amount of motion artifact noise. It can also be seen that the motion artifact noise is indeed difficult to completely filter out and can only be suppressed as much as possible to a certain extent. From the perspective of convergence, as shown in Figure 9, the BNLMS algorithm has the fastest convergence speed, the NLMS algorithm is the second, and the LMS algorithm is the worst. This scheme comprehensively considers the constraints of multiple factors and finally adopts the 8×8 division of the block to be coded, divides the block to be coded into 4 8×8 modes, and performs corresponding motion vector direction analysis on the 4 submodules obtained by the 8×8 mode division.

Select 10 embedded image data in the walking action state, filter these 10 sets of data with the above algorithm, and then, calculate the SNR value and MSE value of these 10 embedded images. The calculation result is shown in Figure 10. It can be seen from the figure that in the walking state, the SNR and MSE values of these 10 embedded images have been improved after these algorithms are processed,

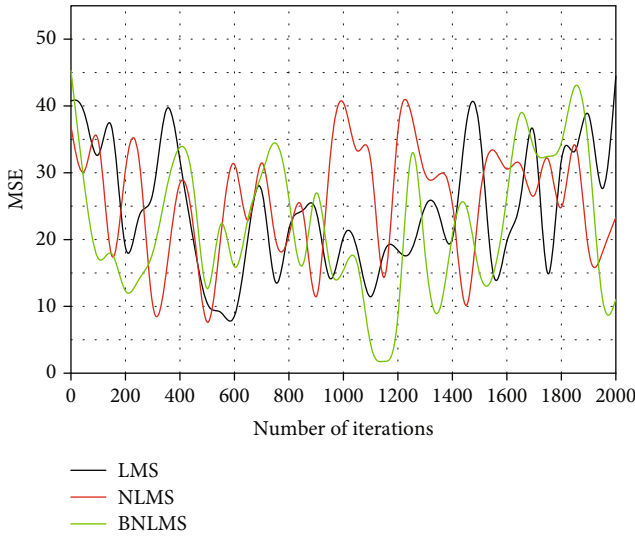


FIGURE 9: Coding curve.

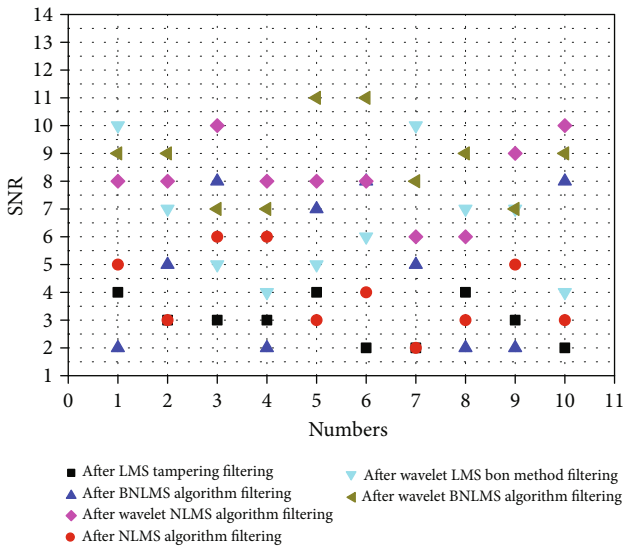


FIGURE 10: Comparison of encoding SNR after filtering by various algorithms.

and the wavelet-based adaptive algorithm is compared with the other three algorithms; the effect of improving the SNR value and MSE value is more obvious.

As shown in Figure 11, through the test results, we can find that the improved UMHexagonS algorithm is good. Since most video sequences move in the horizontal direction more than the vertical direction, the video sequences adopted in this article are biased towards the horizontal direction, so the effect is not obvious.

The test results show that when the signal-to-noise ratio and bit rate change are small, the improved motion estimation time and encoding time are reduced, and the reduction range increases with the increase of the intensity of the motion. At the same time, the code stream analysis software elsewhere streams eye tools that are used to analyze the video

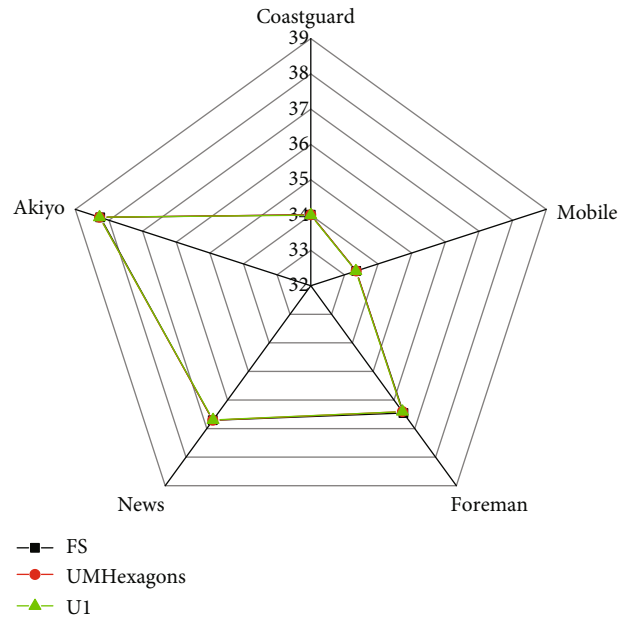


FIGURE 11: Comparison of SNR between each algorithm and the optimized cross template algorithm.

quality before and after optimization. The subjective difference of the video quality before and after optimization is very small, and the optimized target is achieved.

4. Conclusion

As an important part of the video system, video coding can compress video information to reduce the occupation of bandwidth and storage equipment and ensure that the video is spread under the current limited bandwidth and storage space. The coding algorithm is the core of video coding. A good coding algorithm can effectively reduce computational complexity and reduce coding time, which is very important for the improvement of video system performance. Based on the understanding of the key technologies of motion estimation, we tried to modify the adaptive filtering algorithm part and tested it. The test results show that the improved method is feasible; the video data acquisition is controlled by writing a script file, and the algorithm is improved before and after being controlled. The size of the memory occupied by video files, we found that the improved encoding algorithm has higher encoding efficiency and can reduce the storage space occupied by the video.

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 that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] C. Hu, D. Li, Z. Sun, N. Zhang, and J. Lei, "Region-based trilateral filter for depth video coding," *International Journal of Embedded Systems*, vol. 11, no. 2, pp. 163–169, 2019.
- [2] T. Shanableh, "Data embedding in high efficiency video coding (HEVC) videos by modifying the partitioning of coding units," *IET Image Processing*, vol. 13, no. 11, pp. 1909–1913, 2019.
- [3] H. L. Nyo and A. W. Oo, "Secure data transmission of video steganography using Arnold scrambling and DWT," *International Journal of Computer Network and Information Security(IJCNIS)*, vol. 11, no. 6, pp. 45–53, 2019.
- [4] F. Khelifi, T. Brahimi, J. Han, and X. Li, "Secure and privacy-preserving data sharing in the cloud based on lossless image coding," *Signal Processing*, vol. 148, pp. 91–101, 2018.
- [5] E. S. da Silva and H. Pedrini, "Embedded hypercube graph applied to image analysis problems," *Journal of signal processing systems for signal, image, and video technology*, vol. 88, no. 3, pp. 453–462, 2017.
- [6] G. Botella and C. Garcia, "Real-time motion estimation for image and video processing applications," *Journal of Real-Time Image Processing*, vol. 11, no. 4, pp. 625–631, 2016.
- [7] T. L. da Silveira, F. M. Bayer, R. J. Cintra, S. Kulasekera, A. Madanayake, and A. J. Kozakevicius, "An orthogonal 16-point approximate DCT for image and video compression," *Multidimensional Systems and Signal Processing*, vol. 27, no. 1, pp. 87–104, 2016.
- [8] W. Wang and J. Zhao, "Hiding depth information in compressed 2D image/video using reversible watermarking," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4285–4303, 2016.
- [9] Z. Shu, X.-j. Wu, and C. Hu, "Structure preserving sparse coding for data representation," *Neural Processing Letters*, vol. 48, no. 3, pp. 1705–1719, 2018.
- [10] Y. Tew, K. S. Wong, R. C.-W. Phan, and K. N. Ngan, "Multi-layer authentication scheme for HEVC video based on embedded statistics," *Journal of visual communication & image representation*, vol. 40, no. 11, pp. 502–515, 2016.
- [11] T. Dutta and H. P. Gupta, "A robust watermarking framework for high efficiency video coding (HEVC) - encoded video with blind extraction process," *Journal of visual communication & image representation*, vol. 386, pp. 29–44, 2016.
- [12] V. A. Coutinho, R. J. Cintra, F. M. Bayer, P. A. Oliveira, R. S. Oliveira, and A. Madanayake, "Pruned discrete Tchebichef transform approximation for image compression," *Circuits, systems, and signal processing: CSSP*, vol. 37, no. 10, pp. 4363–4383, 2018.
- [13] O. M. Granado, M. O. Martínez-Rach, P. P. Peral, J. O. Gil, and M. P. Malumbres, "Rate control algorithms for non-embedded wavelet-based image coding," *Journal of signal processing systems for signal, image, and video technology*, vol. 68, no. 2, pp. 203–216, 2012.
- [14] Y.-T. Hwang, M.-W. Lyu, and C.-C. Lin, "A low-complexity embedded compression codec design with rate control for high-definition video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 674–687, 2015.
- [15] K. Khurshed, M. Imran, N. Ahmad, and M. O'Nils, "Bi-level video codec for machine vision embedded applications," *Electronics and Electrical Engineering*, vol. 19, no. 8, pp. 93–96, 2013.
- [16] P. H. N. De With, M. J. H. Loomans, and C. J. Koeleman, "Low-complexity wavelet-based scalable image & video coding for home-use surveillance," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 507–515, 2011.
- [17] C. S. Lin, W. J. Yang, and C. W. Su, "FITD: fast intra transcoding from H.264/AVC to high efficiency video coding based on DCT coefficients and prediction modes," *Journal of visual communication & image representation*, vol. 38, no. 7, pp. 130–140, 2016.
- [18] H. Yviquel, A. Sanchez, P. Jääskeläinen, J. Takala, M. Raulet, and E. Casseau, "Embedded multi-core systems dedicated to dynamic dataflow programs," *Journal of signal processing systems for signal, image, and video technology*, vol. 80, no. 1, pp. 121–136, 2015.
- [19] M. U. Khan, M. Shafique, and J. Henkel, "Power-efficient workload balancing for video applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2089–2102, 2016.
- [20] D. Li and S. Xueya, "Development of wireless mobile video surveillance on windows mobile using DirectShow technology," *Journal of Computational and Theoretical Nanoscience*, vol. 14, no. 7, pp. 3163–3169, 2017.
- [21] A. H. Saad and M. Z. Abdullah, "High-speed implementation of fractal image compression in low cost FPGA," *Microprocessors and microsystems*, vol. 47, no. 10, pp. 429–440, 2016.
- [22] M. Grellert, B. Zatt, M. Shafique, S. Bampi, and J. Henkel, "Complexity control of HEVC encoders targeting real-time constraints," *Journal of Real-Time Image Processing*, vol. 13, no. 1, pp. 5–24, 2017.
- [23] H. A. Ilgin and L. F. Chaparro, "Low bit rate video coding using DCT-based fast decimation/interpolation and embedded zerotree coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 833–844, 2007.