Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 11, No. 2, May, Pp. 193-204

Performance evaluation of motion estimation and compensation algorithms in SNR scalable video encoding

Agus Purwadi*[1], Hendra Yufit Riskiawan[1], Agus Hariyanto[1], Nugroho Setyo Wibowo[1], Rani Purbaningtyas[1]
Department of Information Technology, Politeknik Negeri Jember, Indonesia[1]

Article Info

Keywords:
Video Coding, SNR Scalability, Full Search, Three Step Search

Article history:
Received: August 07, 2025
Accepted: December 25, 2025
Published: May 01, 2026

Cite:
A. P. Purwadi, H. Y. Riskiawan, A. Hariyanto, N. S. Wibowo, and R. Purbaningtyas, "Performance Evaluation of Motion Estimation and Compensation Algorithms in SNR Scalable Video Encoding", *KINETIK*, vol. 11, no. 2, May. 2026
https://doi.org/10.22219/kinetik.v11i2.2466

*Corresponding author.
Agus Purwadi
E-mail address:
agus_purwadi@polije.ac.id

Abstract

Motion estimation is the sequential determination of the direction of motion of an object in a video. The movement of an object is denoted by the term motion vector. Between the current and reference frames, motion vectors can signify shift points. The SAD (Sum of Absolute Different) block matching technique is fundamentally dependent on the assessment of an object's motion. This study proposes a hybrid approach that integrates the Three-Step Search (TSS) and Full Search (FS) algorithms. This integration aims to design a block matching algorithm that is applied to video encoding using signal-to-noise ratio (SNR) scalability. From this design, the study aims to obtain the performance results and evaluate the motion estimation process using both the TSS and FS algorithms for performance comparison in SNR scalability video encoding, in terms of video frame quality, bit rate, and PSNR, based on the average comparison of the two algorithms. Based on the experimental results, the FS algorithm achieved a total BD-PSNR of 0.22 dB with an efficiency rate of 12.45%, whereas the TSS algorithm achieved a total BD-PSNR of 0.18 dB and an efficiency rate of 7.6%. Therefore, the FS algorithm demonstrates superior performance compared to the proposed TSS algorithm in video transmission with SNR scalability.

1. Introduction

Motion estimation is a crucial component in video delivery and encoding. Various video processing applications, such as super-resolution, video coding, and video restoration, rely heavily on accurate motion estimation to represent temporal changes between frames. In video coding, motion estimation is used to identify and model the movement of objects over time, which is represented by a motion vector. This motion vector describes the displacement of a point or block of pixels between the current frame and a reference frame, thus reflecting the dynamics of moving objects, changes in viewpoint, and camera movement.

In general, motion estimation techniques can be classified into two main categories: pixel-based and block-based techniques [1],[2],[3]. One of the most widely used methods is the block matching technique, which is an area-based method. This method is widely known for its simplicity of implementation and flexibility in settings. In block matching, the motion vector search process is usually limited to a certain area around the reference block position, which is expressed in the range [±bdx, ±bdy], with the values of *bdx* and *bdy* generally set equal to *bd*. The determination of the BD value is influenced by image resolution, motion characteristics in the video, and coding conditions, both online and offline.

To improve the efficiency and performance of block matching algorithms, various search methods have been developed, including two-dimensional logarithmic search (TDL), three-step search algorithm (TSS), cross-search algorithm (CSA), and one-time search algorithm (OTA) [4],[5],[6].

Despite the widespread use of block matching techniques in video coding systems, they are typically applied only to non-scalable video coding. However, the need for video coding that adapts to network and receiving device conditions is increasing. In practice, only a small number of video coding system providers offer Signal-to-Noise Ratio (SNR) scalability [7],[8],[9],[10]. Conventional single-layer video coding produces only one video output at a specific quality, thus being unable to flexibly adapt to variations in network bandwidth or user quality requirements. This limitation leads to decreased transmission efficiency and quality of service, especially in dynamic network environments.

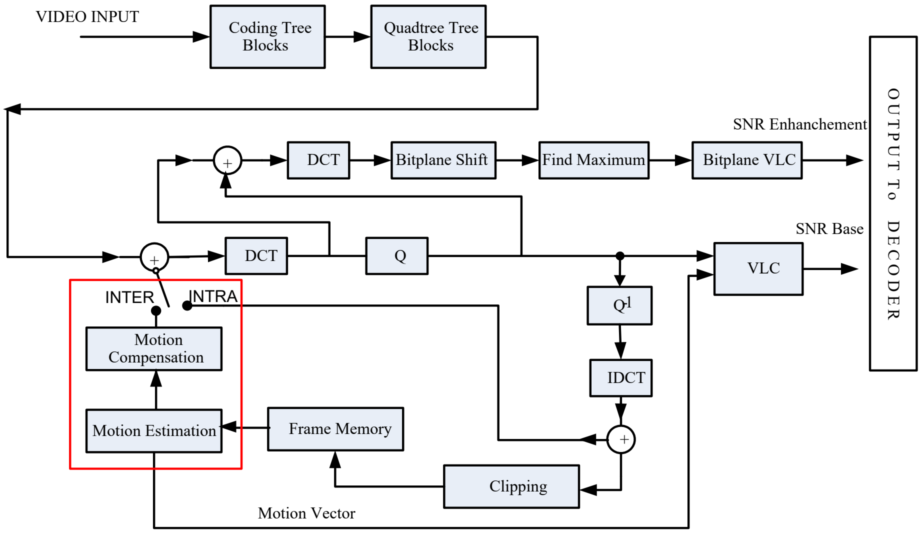
To address these issues, this study proposes a block-matching algorithm-based video coding method that integrates the three-step search (TSS) and full search (FS) algorithms [11],[12],[13],[14]. This method is implemented within a video coding framework with SNR scalability, allowing the creation of a base layer and one or more enhancement layers. With this approach, variations in video frame quality can be generated based on the SNR scalability level and the mean square error value of the obtained motion vectors.

The system performance is evaluated by comparing the average quality between coding schemes with one level of scalability and multiple levels of scalability. The proposed SNR-scalable video coding is expected to overcome the limitations of conventional video coding by providing flexible video output that can be adjusted to network conditions. Thus, users can select video output at the base layer or enhancement layer according to network capacity and quality requirements, significantly improving transmission efficiency and user experience.

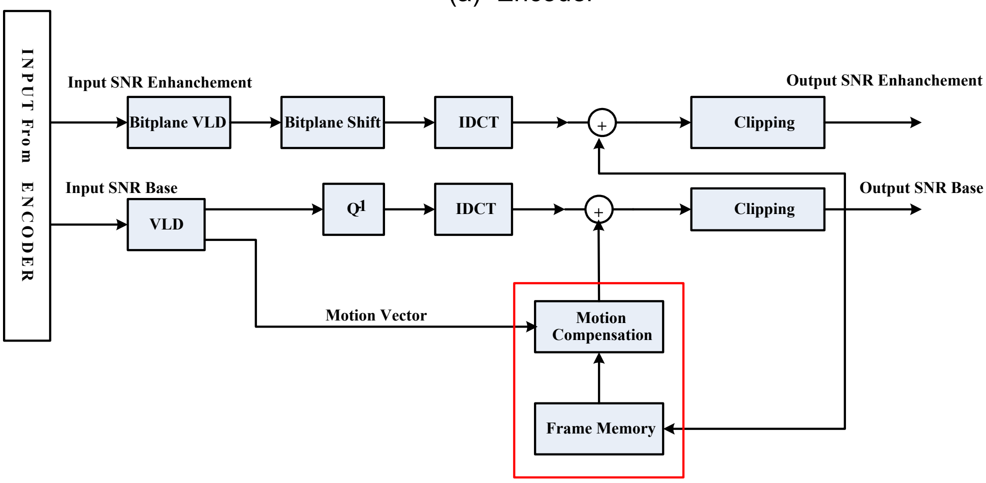
This paper is organized as follows: the first section presents the introduction; the second describes the research method; the third discusses the results and discussion; and the fourth presents the conclusion.

2. Research Design and Methods

A new architecture for scalable video inter-coding based on the High Efficiency Video Coding (HEVC) system is proposed, as depicted in Figure 1. This architecture is designed to be scalable with a base layer and enhancement layers in the experiment in this study. The system obtains performance results from two video coding approaches: the modified motion estimation and compensation full search (FS) method and the block matching three-step search (TSS) approach.



(a) Encoder



(b) Decoder

Figure 1.SNR Scalability Video Coding Encoder-Decoder Block Diagram, The Part Inside the Red Box Is the One Modeled with the TSS Algorithm

2.1 Modeling a Full Search Algorithm

The Full Search Algorithm (FSA) was first used to estimate motion based on blocks. The FSA method precisely checks all search point locations within the lookup region. The optimal FSA can find highly matching positions if the search interval is accurately determined. However, if the search interval in both directions is L and the step size is one pixel, and if the search range is square, the total number of displacements required to determine the motion vector for each block in a frame is $(2L+1)^2$. This is hard to do, especially when the search window is large.

The number of blocks to be searched is $(2\mathrm{L}+1)^2$. The dimension of the pixels is $b \times b$, and the maximum movement of the motion vector is represented by ±w pixels in both vertical and horizontal directions. Figure 2 illustrates the fundamental concept underlying the block matching method, as referenced in [15], [16],[17].

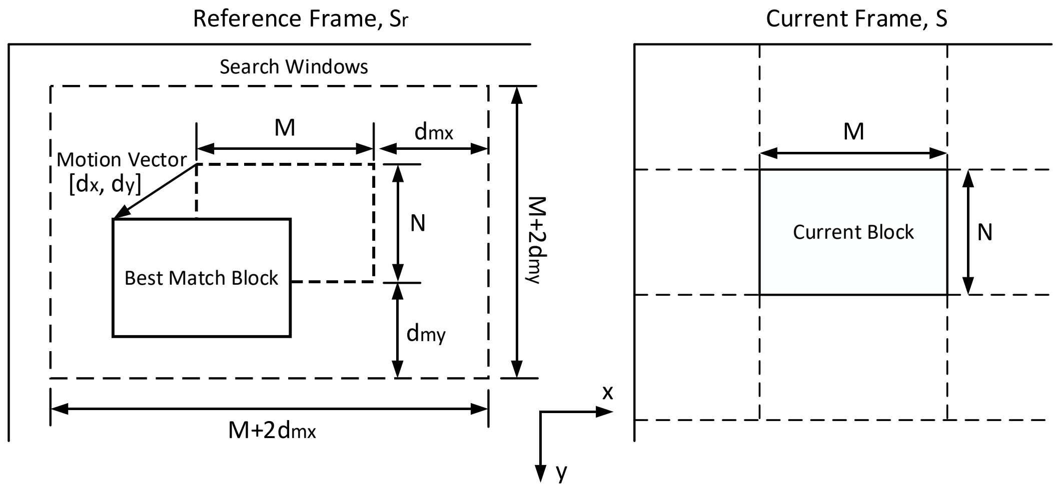


Figure 2. Motion Estimation Through Block Matching

Cross-correlation, mean squared error, and mean absolute error are applied as primary assessment parameters in the analysis. The correlation levels should be multiplied to achieve satisfactory results with two block adjustments in the CCF method. Because CCF is believed to yield suboptimal performance in motion tracking, particularly under low values of the constant L, the implementation utilizes mean square error and mean absolute error instead. Equations 1, 2 and 3 indicate that the functions quantify mean absolute error and mean squared error, while the equation calculates PSNR, where N represents the pixel count of the frame and 255 denotes its 8-bit resolution [18],[19].\

$$\mathrm{MSE}(\mathrm{i},\mathrm{j}) = \frac{1}{\mathrm{N}^2}\sum_{\mathrm{m}=1}^{\mathrm{N}}\sum_{\mathrm{n}=1}^{\mathrm{N}}\big(\mathrm{f}(\mathrm{m},\mathrm{n}) - \mathrm{g}(\mathrm{m}+\mathrm{i},\mathrm{n}+\mathrm{j})\big)^2, \qquad -\mathrm{L} \le \mathrm{I},\mathrm{j} \le \mathrm{L} \tag{1}$$

$$\mathrm{MAE}(\mathrm{I},\mathrm{j}) = \frac{1}{\mathrm{N}^2}\sum_{\mathrm{m}=1}^{\mathrm{N}}\sum_{\mathrm{n}=1}^{\mathrm{N}}|\mathrm{f}(\mathrm{m},\mathrm{n}) - \mathrm{g}(\mathrm{m}+\mathrm{I},\mathrm{n}+\mathrm{j})|, \qquad -\mathrm{L} \le \mathrm{I},\mathrm{j} \le \mathrm{L} \tag{2}$$

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{255^2}{\frac{1}{\mathrm{N}^2}\sum_{\mathrm{m}=1}^{\mathrm{N}}\sum_{\mathrm{n}=1}^{\mathrm{N}}(\mathrm{f}(\mathrm{m},\mathrm{n}) - \mathrm{g}(\mathrm{m}+\mathrm{i},\mathrm{n}+\mathrm{j}))^2}\right] \tag{3}$$

$f(m,n)$ denotes the intensity values within the current block, encompassing an area of N^2 pixels situated at coordinates (m,n), while $g(m+i,n+j)$ signifies a block variable from the previous video frame, relocated to the coordinates $(m+i,n+j)$. Within the optimal matching approach, the locations $i = a$ and $j = b$ are specified such that the motion vector $MV(a,b)$ represents the shift of each pixel inside a block. An exhaustive search is conducted to identify the optimal match, with search parameters defined as $(2L + 1)^2$, where the typical value of L is 7 pixels [20]. The processing load is reduced by using the MAE measurement type, which subsequently became the industry standard video codec. Each N^2 block continuously executes $(2L + 1)^2$ iterations, evaluating every step through $2N^2$ computational processes of addition and subtraction. The PSNR calculation uses N to denote the total pixel count in a frame, and the maximum possible value under 8-bit resolution is 255.

2.2 Modeling a Three-Step Search Algorithm

The proposed approach for block matching is the three-step search. As can be seen in Figure 4, each change results in a twofold reduction in the search area. The three steps of the search are completed when the search radius equals one. The uniformity it brings to both the software and hardware implementations is a benefit of this technique, as it requires a constant number of iterations for each operation. The matching criteria are determined by the Block Distortion Measure (BDM) equation, which is shown in Equation 4 [21],[22],[23].

$$BDM(i,j) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(s(x,y) - s_r(x+i, y+j)) \tag{4}$$

It operates as a function to determine the BDM through $g(.)$, with i and j representing the offsets of the considered motion vector at that time, and x and y denoting the coordinate values relative to the current block. The motion vector is selected as the variable that minimizes the Block Distortion Measure (BDM) for the current block [24],[25]:

$$d = \left[d_x, d_y\right]^T = arg\left(\min_{\forall(i,j)}(BDM(i,j))\right) \tag{5}$$

The implementation steps of the three-step search algorithm can be described through the algorithm and the block diagram shown in Figure 3.

Then, the implementation of the three-step search algorithm for SNR scalability proceeds through the following steps:

a. Set up the parameters used for searching as $r = 2k - 1$ where $k = [log2(dm)]$;
b. Establish checkpoints for $\Gamma = \{[0,0], [\pm r, \pm r], [0, \pm r], [\pm r, 0]\}$;
c. The Block Distortion Measure (BDM) is evaluated along with the SNR scalability at all nine candidate positions, and the point on the lattice exhibiting the lowest combined BDM and SNR scalability value is selected: $d' = arg\left(\min_{(i,j)\in\Gamma}(BDM(i,j))\right)$;
d. Adjust $r = \frac{r}{2}$;
e. If $r < 1$, then $d = d'$; stop. Otherwise, update the search location: $\Gamma = \{[0,0], [\pm r, \pm r], [0, \pm r], [\pm r, 0]\}$, and return to step (c).

The Three-Step Search technique can be computed using Equation 6 [26],[27]:

$$C_{TSS} = 3IJF(8k+1) \tag{6}$$

The system's complexity is denoted by C, the m-th matrix by I, the n-th matrix by J, the frame rate (f/s) by F, and the number of steps employed by k. This model of the Three-Step Search (TSS) algorithm demonstrates its convergence process, initialized with a motion vector denoted as $d = [3,2]$, within a 15×15 search region with a diminishing error rate. The TSS method consistently converges for every video frame in stage N. Nevertheless, the convergence pattern varies depending on the error's shape and rate, as illustrated in Figure 4.

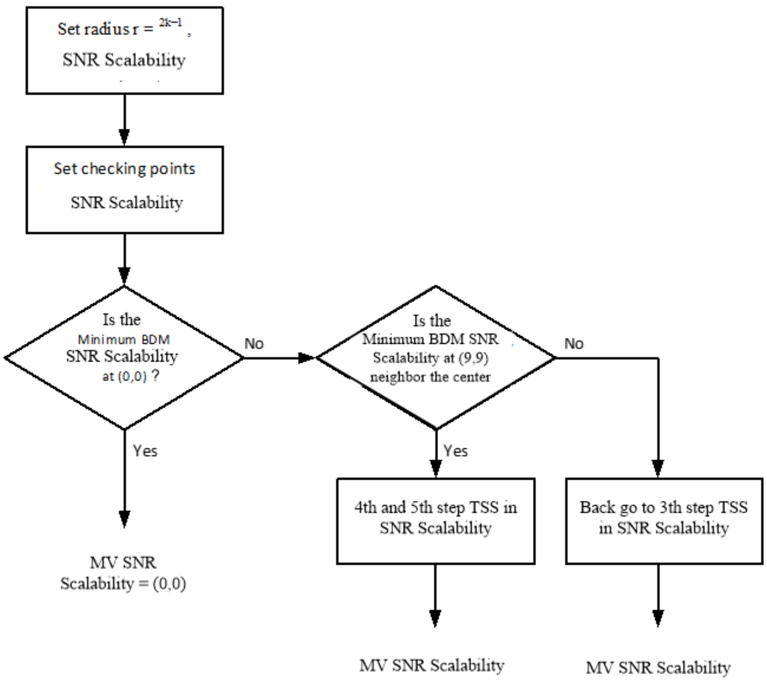


Figure 3. Flowchart for the TSS Algorithm

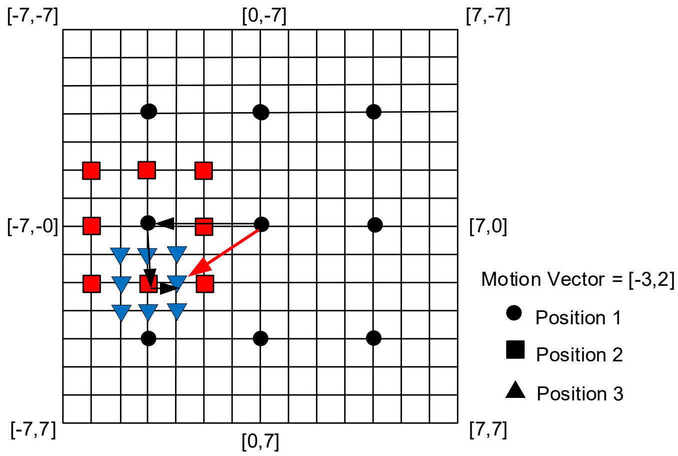


Figure 4. The Three Step Search Structure

2.3 Experimental Models

In this part, the inter-mode SNR scalability of HEVC video coding is implemented using the FS and TSS algorithms to simulate the motion estimation process. The performance of these two motion estimation methods in scaling the SNR of HEVC video coding is evaluated by comparing them side by side. Evaluations are conducted on the PSNR and bitrate performance metrics for both motion estimation method models, specifically the Three-Step Search (TSS) and Full Search (FS), which are designed to accommodate various video media.The results obtained illustrate the performance of SNR scalability in the HEVC video encoder using the two motion estimation algorithms, from which it can be concluded that the proposed technique improves the quality and effectiveness of HEVC video encoded with SNR scalability by applying motion estimation algorithms designed to optimize performance.

Figure 5 presents the video test case applied during the simulation process. For MATLAB and Excel analysis, simulations were run on a Windows 10 PC powered by an Intel (R) Core (TM) i7-67000T processor @ 2.81 GHz. The Y-component, representing luminance, was analyzed over 100 frames at 30 fps to assess SNR scalability [28], [29], [30].

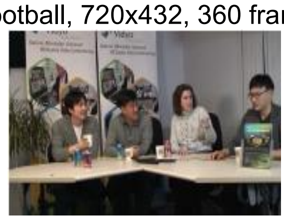
Akiyo, 352×288, 300 frame

Bus, 352×288, 150 frame

Flower Garden, 720x432, 360 frame

Football, 720x432, 360 frame

Shields, 1280x720, 504 frame



Four People, 1280x720, 600 frame

Figure 5. Video Test

Cite: A. P. Purwadi, H. Y. Riskiawan, A. Hariyanto, N. S. Wibowo, and R. Purbaningtyas, "Performance Evaluation of Motion Estimation and Compensation Algorithms in SNR Scalable Video Encoding", KINETIK, vol. 11, no. 2, May. 2026 https://doi.org/10.22219/kinetik.v11i2.2466

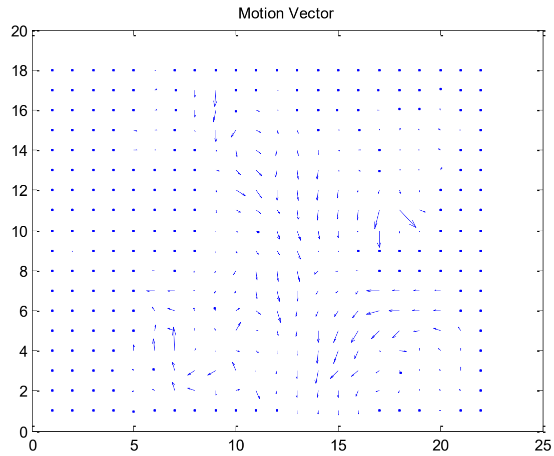


Figure 6. Motion Vector Field on the Akiyo Sequence of 3-4 Frame Pairs with a Time Interval of 1.3740 Seconds at 30 fps

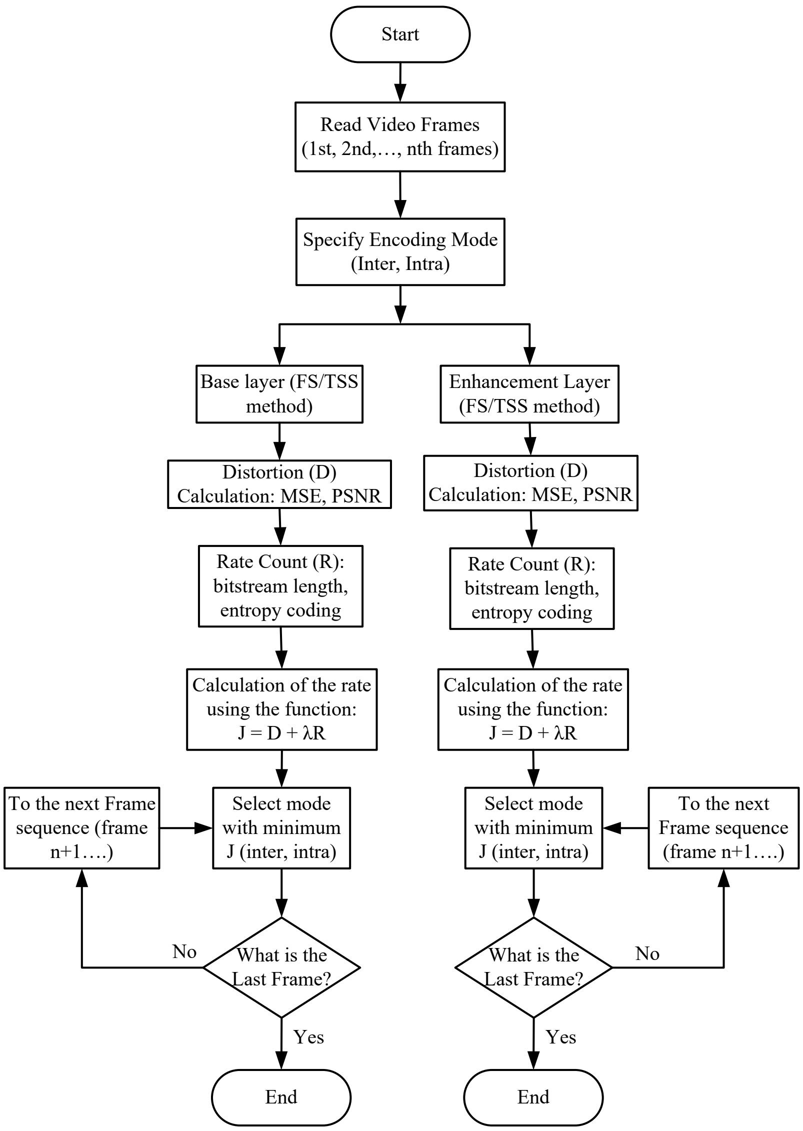


Figure 7. System Experiment Flowchart

Figure 7 illustrates the experimental process and testing of video coding based on Rate-Distortion Optimization (RDO) with the implementation of the Full Search (FS) and Three Step Search (TSS) algorithms on the base layer and enhancement layer. The process begins by reading the video frame sequence (frames 1 to n); then the system determines the coding mode to be evaluated, namely intra or inter. For each frame, the distortion (D) is calculated using the MSE and PSNR metrics. In the next stage, motion estimation and coding are performed using the FS/TSS method on each layer. Next, the bit rate (R) is calculated based on the bitstream length and the entropy coding results. The distortion and bit rate values are then combined in the RDO cost function, namely $J = D + \lambda R$, to determine the coding mode (intra or inter) with the minimum cost value on each frame by selecting the smallest J value. D represents the distortion, namely the difference between the original frame and the reconstructed frame, measured using MSE and PSNR. λ is a parameter that regulates the balance between video quality and bit rate. This function ensures the selection of an encoding mode that balances visual quality and bit rate efficiency on each frame. Once the best mode is selected, the system checks whether the processed frame is the last frame. If not, the process continues to the next frame; if so, the encoding process ends. This flowchart shows that the encoding mode selection is performed iteratively and optimally on each frame, considering the balance between visual quality (distortion) and bit rate efficiency at both encoding layers.

3. Results and Discussion

The results of a simulation experiment for motion estimation and compensation in video coding using the Gradient Descent method, an algorithm that numerically estimates the minimum value of a function, are shown in Figure 8 for the Y component (luminance) of the video test sequence. Copies are created for the target frame, anchor frame, motion vector, and predicted frame through this process.

Figure 6 shows that the PSNR value changes depending on the number of motion vector directions in the video test sequence. The simulation results show that the estimated and compensated PSNR values for frames 3–4 of the Akiyo sequence is 32.9674 dB for the predicted frame.

The simulation results demonstrate the performance of the video coding system with SNR scalability, utilizing the FS and TSS algorithms for motion estimation. The objective of this analysis is to derive quantitative metrics of video frame quality by computing the average values of PSNR, MSE, CPU time, BD-PSNR, and BD-rate across different scalability levels and layers, as illustrated in Table 1. The results of simulations on motion estimation using the FS method with SNR-scalable video coding show that the average PSNR value for all test videos is 32.9 dB for the base layer and 66.6 dB for the enhancement layer. The average mean square error (MSE) for the base layer is 0.9, while for the enhancement layer it is 0.88. The base layer has an average bit rate of 14,068.6 Kbps, while the enhancement layer has 33,368.2 Kbps. The computation time for the base layer is 9,095.2 seconds, while for the enhancement layer it is 9,035.2 seconds. The BD-PSNR gain for all test videos is 0.22 dB, and the BD-rate efficiency is 12.45%. These results vary across all test parameters, reflecting the performance of the video coding system with SNR scalability. The results are then compared with the TSS algorithm, which serves as a benchmark to evaluate the performance of the system.

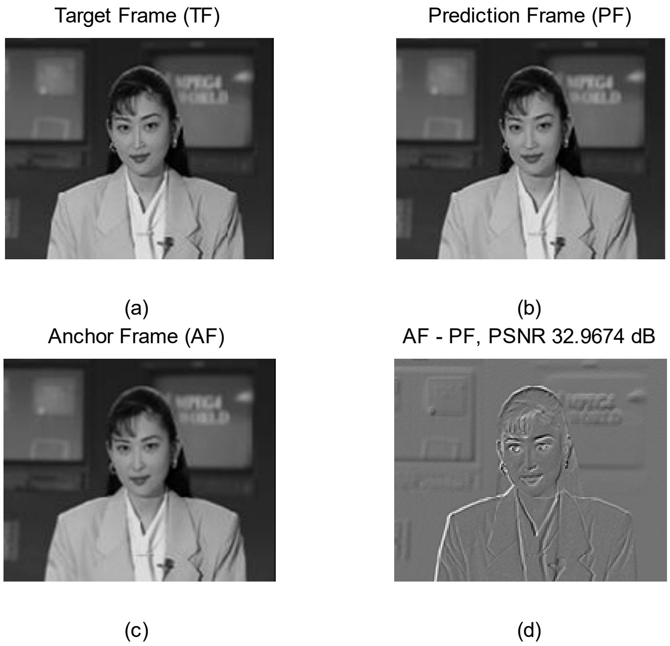


Figure 8. Display (a) Target Frame, (b) Prediction Frame, (c) Anchor Frame, (d) PSNR Value (anchor frame – prediction frame)

Table 1. Experimental Findings of Full Search (FS)

Video	Mean PSNR (dB)		Mean MSE		Mean Bit Rate (Kbps)		CPU Time (s)		BD PSNR (dB)	BD-rate (%)
	Base layer	Enhancement Layer	Base Layer	Enhancement Layer	Base Layer	Enhancement Layer	Base Layer	Enhancement Layer		
Akiyo	36.6	73.1	0.9	0.8	268.8	537.5	1,827.3	1,771.4	0.3	78.8
Bus	32.5	64.9	0.9	0.9	5,637.9	11,274.6	2,921.3	1,540.2	0.2	-1
Football	32.8	65.4	0.9	0.9	21,039.7	42,081.9	7,381.6	7,167.1	0.2	-1
Flower Garden	33.1	66.4	0.9	0.9	28,319.5	56,641.2	14,735.9	16,425.9	0.1	-0.1
Shields	27.8	60.4	0.9	0.9	29,134.5	66,777.3	14,972.6	14,880.5	0.2	-1
Four People	34.7	69.2	0.9	0.9	11,451	22,896.5	12,732.3	12,429.5	0.3	-1
Mean	32.9	66.6	0.9	0.88	14,068.6	33,368.2	9,095.2	9,035.7	0.22	12.45

Table 2 illustrates the motion estimation results obtained using the TSS algorithm applied to SNR-scalable video coding, serving as a comparison with the default system performance of the FS algorithm. The average PSNR value for all test videos is 29.6 dB for the base layer and 59.5 dB for the enhancement layer. The average mean square error (MSE) for both the base and enhancement layers is 0.9. The mean data rate for the base layer is 14,405.5 Kbps, while the enhancement layer is 28,809.6 Kbps. The computation time required for the base layer is 9,108.5 seconds, while for the enhancement layer it is 9,251.8 seconds. The BD-PSNR gain for all test videos is 0.18 dB, and the BD-rate efficiency is 7.6%. These results vary across all test parameters, reflecting the performance of the video coding system with SNR scalability. The results obtained using the TSS algorithm are compared with those of the FS algorithm to evaluate system performance.

Table 2. Experimental Findings of Three-Step Search (TSS)

Video	Mean PSNR (dB)		Mean MSE		Mean Bit Rate (Kbps)		CPU Time (s)		BD PSNR (dB)	BD-rate (%)
	Base layer	Enhancement Layer	Base Layer	Enhancement Layer	Base Layer	Enhancement Layer	Base Layer	Enhancement Layer		
Akiyo	34	68	0.9	0.9	203.2	406.4	1,743.6	1,797	0.2	50.7
Bus	27.3	55.2	0.9	0.9	4,653.7	9,307.6	1,591.3	1,628	0.1	-1
Football	28.5	57.3	0.9	0.9	17,376.6	34,754.3	7,434.4	7,722.1	0.2	-1
Flower Garden	29.6	59.5	0.9	0.9	24,428.3	48,858.6	14,454.2	14,601.4	0.2	-1
Shields	28.3	56.2	0.9	0.9	30,075.1	60,142.4	16,931	13,804.8	0.2	-1
Four People	30.1	60.8	0.9	0.9	9,696.1	19,388.2	12,496.8	15,957.8	0.2	-1
Mean	29.6	59.5	0.9	0.9	14,405.5	28,809.6	9,108.5	9,251.8	0.18	7.6

The experimental results of bit rate and Peak Signal-to-Noise Ratio (PSNR) for all test video sequences using the Full Search (FS) and Three Step Search (TSS) algorithms are presented in Figure 9. It can be observed from the figure that the enhancement layer generated by the FS algorithm consistently achieves the highest PSNR and bit rate across all test sequences. In contrast, the lowest PSNR and bit rate are obtained at the base layer of the TSS algorithm for all video sequences. This performance difference is primarily attributed to the accuracy of motion vector estimation for each frame. The FS algorithm performs an exhaustive search over the entire search window, enabling it to determine more accurate motion vectors. Conversely, the TSS algorithm restricts the search process to only three search steps, resulting in lower computational complexity at the expense of reduced motion estimation accuracy.

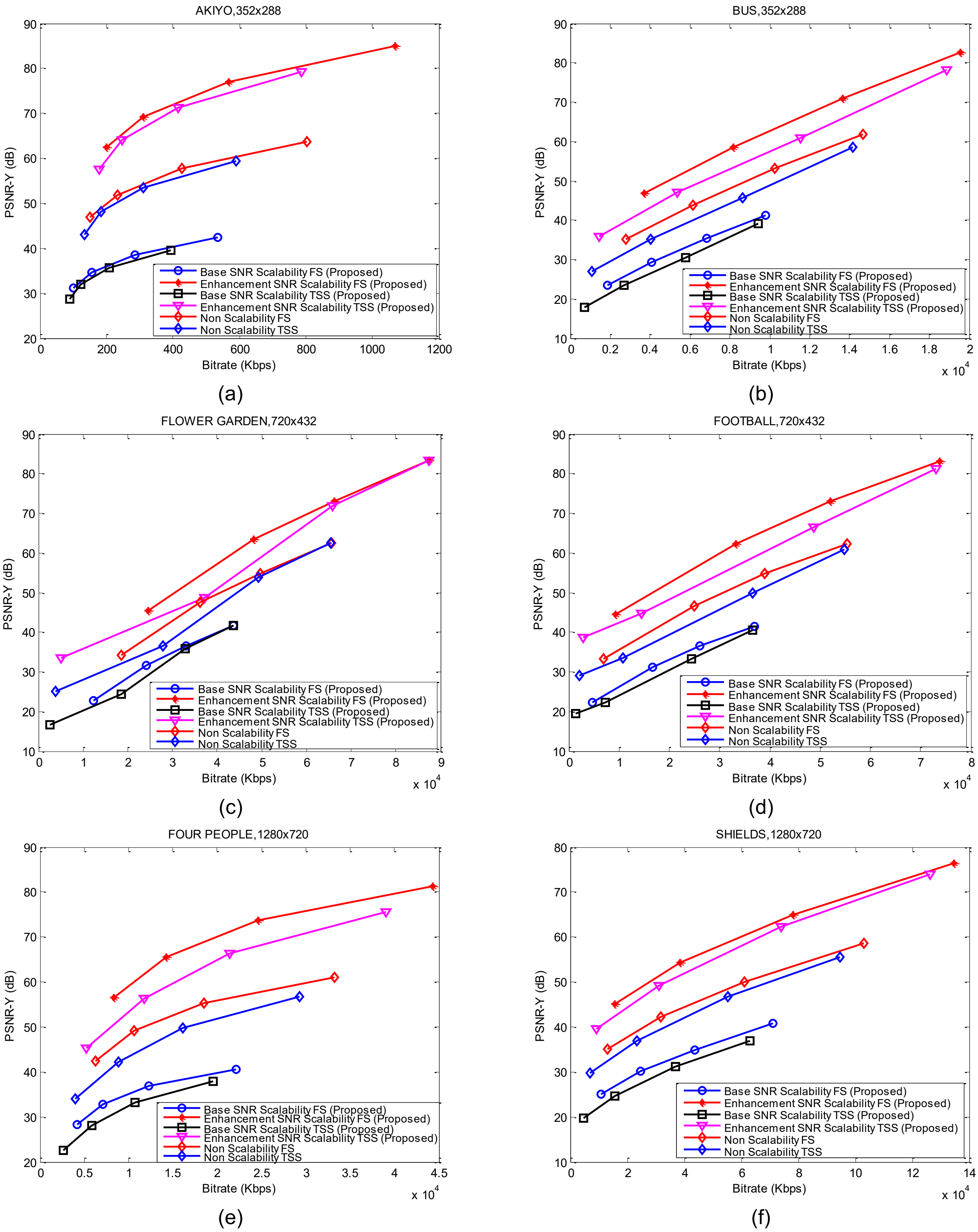
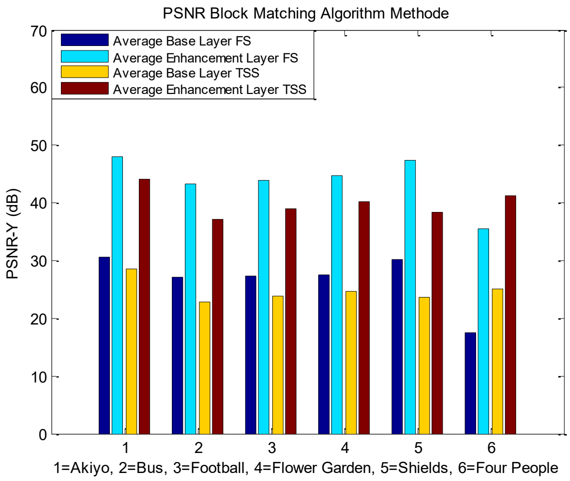


Figure 9. Bitrate Performance SNR Scalability Graph

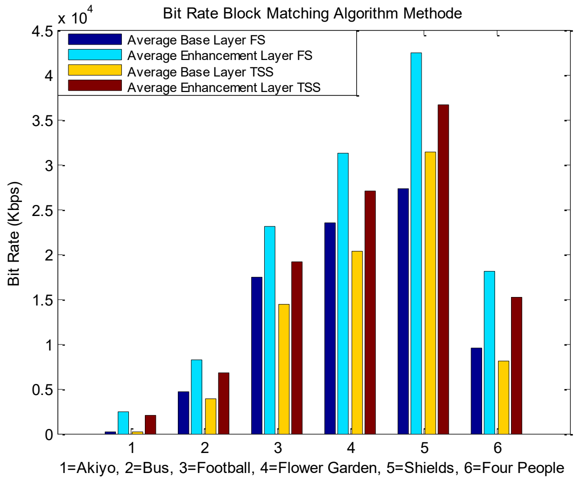
Nevertheless, each algorithm exhibits distinct trade-offs between compression efficiency and computational complexity. When high video quality is required, albeit with a relatively high bit rate and increased bandwidth consumption, the base layer generated by the FS algorithm is more suitable. On the other hand, for applications where bandwidth efficiency is prioritized over reconstruction quality, the enhancement layer produced by the TSS algorithm offers a viable solution due to its relatively low bit rate requirements.

To further evaluate the overall system performance, Figure 8 also compares the proposed scalable video coding system with a non-scalable video coding system. The results indicate that the scalable video coding approach employing FS for both the base and enhancement layers achieves the best overall performance compared to the non-scalable coding schemes using FS and TSS across all test sequences. However, the non-scalable video coding schemes based on FS and TSS still outperform the scalable coding system employing TSS for both the base and enhancement layers.

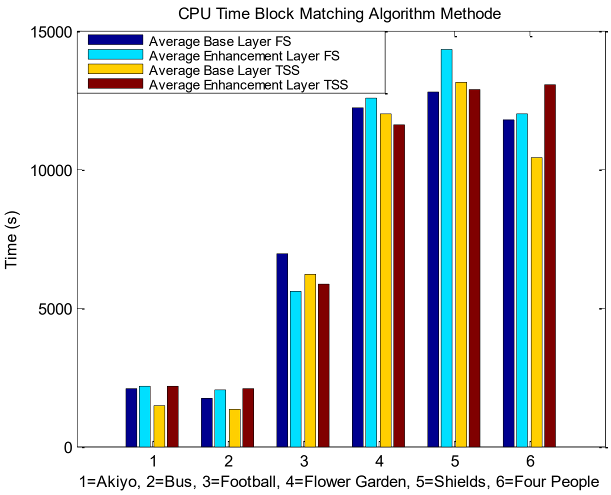
Figure 10 shows a graph of the difference between the Full Search (FS) and Three-Step Search (TSS) algorithms block-matching methods used in video coding with SNR scalability. In terms of PSNR performance, the enhancement layer produced by the FS algorithm records the highest average across most tested video sequences, while the base layer of the TSS algorithm has the lowest average value for most test videos. In terms of bit rate performance, the highest value is on the average enhancement layer of the FS algorithm on the Shields test video, and the smallest is on the average base layer of the FS algorithm for the Akiyo test video. In terms of computational time performance, the average enhancement layer of the FS algorithm for the *Shields* test video has the longest computation time, while the average base layer of the FS algorithm for the *Bus* test video has the shortest computation time.



(a)



(b)



(c)

Figure 10. Full Search versus Three-Step Search in Block-Based Motion Estimation (a) PSNR, (b) Bit Rate, (c) CPU Time

4. Conclusion

According to the findings of the tests and assessments, the FS algorithm continues to perform better than the suggested TSS method, as demonstrated by the enhancement layer, which records a higher average bit rate, a higher average PSNR value, and a higher CPU time. The TSS algorithm's enhancement layer has the greatest average bit rate and PSNR value in the system assessment, after the FS algorithm's enhancement layer. The TSS algorithm has an efficiency rate of 7.6% and a total BD-PSNR value of 0.18 dB, whereas the FS algorithm has an efficiency rate of 12.45% and a total BD-PSNR value of 0.22 dB. Therefore, the FS algorithm outperforms the suggested TSS algorithm in video transmission with SNR scalability coding. The enhancement layers with FS and TSS algorithms can be customized according to user requirements to implement a system that ensures strict performance standards in HEVC video coding through SNR scalability integration.

Notation

$M \times N$	: Original block size video frame
$\forall r^{\rightarrow}$	: Motion vector
$(2L+1)^2$	: Block number
$MSE(i,j)$	: Mean Square Error (i,j)
$MAE(i,j)$	: The Mean Absolute Error at candidate motion vector (i, j)
$MV(a,b)$	: Movement Vector (a,b)
$BDM(i,j)$	: Block Distortion Measure (i,j)
$[ax, ay]$	: Middle position pixel
$\psi(x, y, t_1)$, $\psi(x, y, t_2)$	: Motion between two pictures
$I'(x',y')$	: Present frame
$I(x)$	: Preceding frame
$a^{(t+1)}$	: Iteration motion estimation
$r = {}^{2k-1}$	: Set the search region
Γ	: Pixel configuration checkpoints
d'	: SNR-based scalability value
$(2d_m + 1)^2$	: Motion vector candidates
$[d_x, d_y]^T$	: Minimize the existing block's distortion measure

Acknowledgement

We sincerely thank all individuals and institutions who have supported the completion of this research. Special appreciation is directed to the Information Technology Department of Politeknik Negeri Jember for its continuous support throughout the research process and for providing the necessary funding for this publication.

References

[1] S. Chen, L. Huang, Z. Zan, X. Zeng and Y. Fan, "An Interpolation-Free Fractional Motion Estimation Algorithm and Hardware Implementation for VVC," in *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 33, no. 2, pp. 395-407, Feb. 2025. https://doi.org/10.1109/TVLSI.2024.3455374

[2] L. Jia, C. -Y. Tsui, O. C. Au and K. Jia, "A Low-Power Motion Estimation Architecture for HEVC Based on a New Sum of Absolute Difference Computation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 243-255, Jan. 2020. https://doi.org/10.1109/TCSVT.2018.2890204

[3] Dayong Wang, Yu Sun, Ce Zhu, Weisheng Li, Frederic Dufaux, "Fast Depth and Inter Mode Prediction for Quality Scalable High Efficiency Video Coding", *IEEE Transactions on Multimedia*, Vol. 22, No. 4, April 2020. https://doi.org/10.1109/TMM.2019.2937240

[4] I. Storch, L. Agostini, B. Zatt, S. Bampi and D. Palomino, "FastInter360: A Fast Inter Mode Decision for HEVC 360 Video Coding," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 3235-3249, May 2022. https://doi.org/10.1109/TCSVT.2021.3096752

[5] Y. Lou, H. Lin, Q. Chen, M. Xing, S. Zhou and G. -C. Sun, "2-D Autofocus for High-Squint SAR Based on Affine Coordinate Back-Projection Algorithm," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-17, 2024, Art no. 5228217. https://doi.org/10.1109/TGRS.2024.3493095

[6] A. Purwadi, Wirawan, and Suwadi, "Improved HEVC Video Encoding Quality with Multi Scalability Techniques," in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia: IEEE*, Dec. 2021, pp. 306–311. https://doi.org/10.1109/ISRITI54043.2021.9702868

[7] M. Wien, "High Efficiency Video Coding", in *Signals and Communication Technology. Berlin, Heidelberg: Springer Berlin Heidelberg*, 2015. https://doi.org/10.1007/978-3-662-44276-0

[8] T. Shinde, "Adaptive Pixel-based Direction Oriented Fast Motion Estimation for Predictive Coding," *2022 Picture Coding Symposium (PCS), San Jose, CA, USA*, 2022, pp. 331-335. https://doi.org/10.1109/PCS56426.2022.10018044

[9] R. Mukherjee and A. Puniyani, "A Fast Adaptive Algorithm for Sub-Pixel Motion Estimation," *2022 IEEE Region 10 Symposium (TENSYMP), Mumbai, India*, 2022, pp. 1-5. https://doi.org/10.1109/TENSYMP54529.2022.9864538

[10] T. Yan, I. -H. Ra, H. Wen, M. -H. Weng, Q. Zhang and Y. Che, "CTU Layer Rate Control Algorithm in Scene Change Video for Free-Viewpoint Video," in *IEEE Access*, vol. 8, pp. 24549-24560, 2020. https://doi.org/10.1109/ACCESS.2020.2970063
[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circuits Syst. Video Technol*., vol. 22, no. 12, pp. 1649–1668, Dec. 2012. https://doi.org/10.1109/TCSVT.2012.2221191
[12] "High efficiency video coding", ITU-T Recomendation H.265, 2019
[13] J. -P. Hong, Z. -H. Dong, X. Zhang and N. -N. Song, "Optimization method of integer pixel motion estimation algorithm based on HEVC," *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), Xi'an, China*, 2023, pp. 1378-1382. https://doi.org/10.1109/ICSP58490.2023.10248819
[14] J. Li, M. Zhang, K. Niu, Y. Zhang, and X. Yang, “A HEVC Video Steganalysis Method Using the Optimality of Motion Vector Prediction”, *Computers, Materials and Continua*, Vol. 79, No. 2, pp. 2085-2103, ISSN 1546-2218, 2024. https://doi.org/10.32604/cmc.2024.048095
[15] V. Sze, M. Budagavi, and G. J. Sullivan, Eds., High Efficiency Video Coding (HEVC): Algorithms and Architectures. in *Integrated Circuits and Systems. Cham: Springer International Publishing*, 2014. https://doi.org/10.1007/978-3-319-06895-4
[16] E. Çetinkaya, H. Amirpour, M. Ghanbari, and C. Timmerer, “CTU depth decision algorithms for HEVC: A survey,” *Signal Process. Image Commun*., vol. 99, p. 116442, Nov. 2021. https://doi.org/10.1016/j.image.2021.116442
[17] S. Jeon, A. O.-A. Antwi, and K. Ryoo, “Efficient Motion Estimation Algorithms for High-Performance HEVC Encoder,” *presented at the Green and Smart Technology 2016*, Dec. 2016, pp. 53–57. https://doi.org/10.14257/ASTL.2016.141.11
[18] P. Nalluri, L. N. Alves, and A. Navarro, “Complexity reduction methods for fast motion estimation in HEVC,” *Signal Process. Image Commun.*, vol. 39, pp. 280–292, Nov. 2015. https://doi.org/10.1016/j.image.2015.09.015
[19] L. R. Duong, B. Li, C. Chen and J. Han, "Multi-Rate Adaptive Transform Coding for Video Compression," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece*, 2023, pp. 1-5. https://doi.org/10.1109/ICASSP49357.2023.10095879
[20] D. Rüfenacht, Novel Motion Anchoring Strategies for Wavelet-based Highly Scalable Video Compression. in *Springer Theses. Singapore: Springer Singapore*, 2018. https://doi.org/10.1007/978-981-10-8225-2
[21] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD curves", *ITU-T SG 16/Q.6 13th VCEG Meeting, Austin, Texas, USA*, document VCEG-M33, 2001
[22] S. Gogoi and R. Peesapati, “Design and Implementation of Gray-Coded Bit-Plane Based Reconfigurable Motion Estimation Architecture Using Binary Content Addressable Memory for Video Encoder”, *IEEE Transactions on Consumer Electronics*, Vol. 68, No. 1, pp. 85-92, 2022. https://doi.org/10.1109/TCE.2021.3139944
[23] Xiph.org Foundation. 2017, “Derf’s test media collection.”
[24] Fraunhofer Heinrich Hertz Institute, “HEVC test model HM-16.18 Documentation”, *2018,*
[25] S. R. Alvar, M. Abdollahzadeh and H. Seyedarabi, "A novel fast search motion estimation algorithm in video coding," *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, 2014*, pp. 934-937. https://doi.org/10.1109/ISIE.2014.6864737
[26] C. -J. Chiu, Y. -C. Tien, K. -T. Feng and P. -H. Tseng, "Intelligent Visual Acuity Estimation System with Hand Motion Recognition," in *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 6226-6239, Dec. 2021. https://doi.org/10.1109/TCYB.2020.2969520
[27] S. Gogoi and R. Peesapati, "Design and Implementation of an Efficient Multi-Pattern Motion Estimation Search Algorithm for HEVC/H.265," in *IEEE Transactions on Consumer Electronics*, vol. 67, no. 4, pp. 319-328, Nov. 2021. https://doi.org/10.1109/TCE.2021.3126670
[28] Agus Purwadi, Suwadi, and Wirawan,” Lagrange Multiplier Selection for Rate–Distortion Optimization in Multi-scalable Video Coding,” *International Journal of Intelligent Engineering and Systems*, Vol.18, No.2, 2025. https://doi.org/10.22266/ijies2025.0331.04
[29] C. Herglotz et al., “The Bjøntegaard Bible Why Your Way of Comparing Video Codecs May Be Wrong”, *IEEE Transactions on Image Processing*, Vol. 33, pp. 987-1001, 2024. https://doi.org/10.1109/TIP.2023.3346695
[30] D. García-Lucas, G. Cebrián-Márquez, A.J. Díaz-Honrubia, T. Mallikarachchi, and P. Cuenca, “A fast full partitioning algorithm for HEVC-to-VVC video transcoding using Bayesian classifiers”, *Journal of Visual Communication and Image Representation*, Vol. 94, 103829, 2023. https://doi.org/10.1016/j.jvcir.2023.103829