Received: May 14, 2024. Revised: June 14, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Implementation of Motion Estimation Algorithms in Multi-scalability to Provide High-efficiency Video Coding

Agus Purwadi[1,2] Suwadi[1]* Wirawan[1] Esa Prakasa[3]

[1]*Electrical Engineering Department, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia*
[2]*Information Technology Department, Politeknik Negeri Jember, Jember, Indonesia*
[3]*Research Center for Data and Information Sciences, Research Organization for Electronics and Informatics, National Research and Innovation Agency (BRIN), Indonesia*
* Corresponding author's Email: suwadi@ee.its.ac.id

Abstract: Motion estimation involves determining the direction of an object's movement, which is crucial in video coding during the transmission process. The motion vector can indicate the shift point between the currently processed frame and the frame used as a processing reference. The sum-of-absolute-difference (SAD) block-matching algorithm relies heavily on estimating object movement. In this research, we propose the integration of three-step search (TSS) and full search (FS) methods in multi-scalable video transmission using high-efficiency video coding (HEVC), applying three scalabilities—spatial, signal-to-noise ratio (SNR), and temporal. This integration aims to increase efficiency and improve quality by employing a block-matching algorithm. With this design, we evaluate the performance of the TSS and FS methods in multi-scalable video coding, obtaining the video frame quality with peak SNR (PSNR) and bit rate efficiency. From the results of experiments using video tests in Standard Definition (SD), Common Intermediate Format (CIF), and High- Definition (HD) formats, the FS algorithm has a total Bjontegaard delta (BD)-PSNR value of 0 dB and an efficiency of 62.4%, while the TSS achieves a total BD-PSNR value of 0.8 dB and an efficiency of 23.6%. Meanwhile, the optimal PSNR and bit rate for the multi-scalability average were found with the FS algorithm enhancement layer and the TSS algorithm enhancement layer.

Keywords: HEVC, SVC, SAD, FS, TSS.

1. Introduction

Motion estimation is crucial for video transmission, particularly in the video processing section. Many video processing applications, including video coding, super-resolution, and video restoration, require movement estimation. In video coding, movement estimation is necessary for determining the sequential movement of objects in a video, as described by the motion vector of each object. In other words, the movement of a point between the current and subsequent frames can be represented by a vector, which can demonstrate any movement issues around the observed structures. We can also observe the correlation between temporal elements, such as a moving object, a shifting viewpoint, or a moving camera.

The motion estimation techniques are conventionally categorized into two groups, namely, pixel- and block-based structures [1]. The approach in this paper is an area-based method known as block matching. The block-matching algorithm is well known for its ease of implementation and simplicity, where search frames are usually limited to an area $[\pm d_{mx}, \pm d_{my}]$, where $d_{mx} = d_{my} = d_m$. The d_m value depends on image resolution, temporal activity, and online or offline encoding. Four techniques can be implemented to develop the block-matching algorithm: (1) two-dimensional logarithmic (TDL) searches, (2) three-step search (TSS) algorithm, (3) cross-search algorithm (CSA), and (4) one-time search algorithm (OTA) [2].

Algorithms for block matching have been widely used in video coding, especially for the full search (FS) algorithm, which almost all standard video coding

uses. However, the FS algorithm is only applied to single scalability coding, and still, no one algorithm offers video coding with complete scalability or multi-scalability. Multi-scalability includes spatial, temporal, and signal-to-noise ratio (SNR) scalability. Each scalability involves a base layer and an enhancement layer built into one video encoder with six output options. This research aims to design video coding that is superior and efficient in bit rate and quality [3 - 6].

Therefore, this research proposes a block-matching algorithm method using the TSS and FS algorithms (widely used in video coding) that are applied to high-efficiency video coding (HEVC) [7] with some output scalability, consisting of SNR, spatial, and temporal scalability using base and enhancement layers [4, 5, 8]. In the proposed multi-scale video coding, the goal is to increase the power and selectivity of motion estimation in video transmission with algorithm variations in motion estimation, with selectivity to obtain a model using a precise and efficient motion-estimation algorithm in video coding with multi-scalability.

In addition, multi-scalable video coding, which can overcome the weakness of single-scalable video coding, needs to be employed to obtain various outputs in video coding. Fig. 1. illustrates video coding with multiple output scalabilities, including spatial, SNR, and temporal scalability. Each scalability function uses a base and enhancement layers with the output of several video formats such as Quarter Common Intermediate Format (QCIF), Common Intermediate Format (CIF) and High Definition (HD). Table 1 summarizes the development of motion estimation and compensation methods and scalability strategies in video coding discussed in this study.

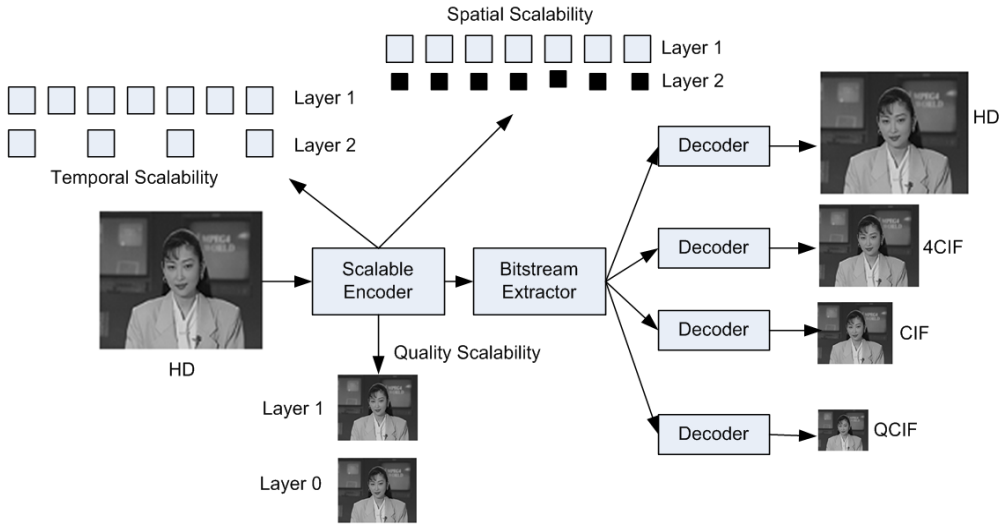


Figure. 1 Implementation of the multi-scalability in transmitting video data

Table 1. Summary of the related motion estimation and scalability methods

Categories	Methods	Domains	Main ideas
Rate distortion Lagrange multiplier	F. Zhang et al. [25]	Based on HEVC	Based on the Lagrange multiplier with single scalability in HEVC
Motion estimation and compensation	Qian Liu et al. [26]	Based on HEVC	Based on the FS, diamond, and TSS algorithms for single scalability in HEVC
	K. Yang et al. [27]	Based on HEVC	Based on HEVC with temporal scalability (single scalability) layer enhancement
	R. Bailleul et al. [28]	Based on HEVC	Based on HEVC with SNR scalability (single scalability) layer enhancement
	Z. Shi et al. [4]	Based on HEVC	Based on HEVC with spatial scalability (single scalability)
	Proposed	**Based on HEVC**	**Based on the selectivity of the FS and TSS algorithms in HEVC with multiple scalability (SNR, spatial, and temporal) in one video coding with various outputs**

The remainder of this article is organized as follows: Section 2 presents relevant work; Section 3 discusses HEVC video coding, followed by our suggested technique in Section 4; the experimental model and results are presented in Section 5. Finally, Section 6 outlines the conclusion.

2. Related works

This section presents a series of coherent and practical explanations of the development of motion estimation and compensation methods as well as scalability strategies in HEVC video coding discussed in this research. Researchers in [26] conducted research based on FS, diamond, and TSS algorithms for single scalability in HEVC, which only provides one output option. Researchers [27] conducted research using HEVC with improved temporal scalability layers. This research is also single-scalable because it has one video output. The research in [28] focuses on HEVC coding with increased SNR scalability layers; it also includes video coding with a single scalability method. The research in [4] focuses on HEVC video coding with spatial scalability, also known as video coding with single scalability. As long as video coding still uses single-scalability, it only has one video output. This video coding is still incomplete for providing video-on-demand services. In this research, we develop a single scalable video code into a multi-scalable video code that provides SNR, spatial, and temporal scalable output. This research also compares the FS and TSS algorithm methods [26], which are applied to HEVC video coding with multi-scalability. This research is expected to provide a solution to obtain video coding with a high level of efficiency.

2.1 HEVC

HEVC is a standard for video encoding generated by the Video Coding Joint Collaborative Team (JCT-VC). It embeds contributions from ISO MPEG and ITU-T VCEG [7, 11, 12]. HEVC is founded on the hybrid video code that uses a block design concept. The video sequence frames are partitioned down into blocks, and intra-prediction is employed for each block. The preceding predictive mode only used samples decoded in the identical frame for video frame references, whereas HEVC uses a block of previously decoded frames. Typically, frames in predictive mode anticipate the motion of objects in a video sequence among frames. The term for this is predictive motion compensation. Intra-predictive mode uses adjacent block spatial redundancy, whereas prediction with motion compensation exploits temporal redundancy between video frames. Using the spatial redundancy of the video frame block, the prediction error is transmitted via linear transformation, scalars in the transformation coefficient's quantization, and entropy coding as a result of the transformation coefficient.

The structure of the block-based video encoder and decoder with fine granularity scalability (FGS) video compression and characteristics related to HEVC are shown in Fig. 5. [13]. In the newly developed block partitioning method, every video frame in HEVC is partitioned to be independent and the same size square block, with each block functioning as the first block partition's quadtree structure's origin in the encoding tree, known as a coding tree block (CTB). The CTB is subdividable along the encoding into a tree structure of multiple code blocks (CBs), and the encoder uses predictive intra-picture compensation to determine the movements in the video frame. The residual structure of the block partition quadtree is used for coding with the predicted residual transformation at the CB level [14 - 16].

2.2 Block matching algorithm in video encoding

Block matching in video frames is the most common video coding technique, and it can predict movement in a sequence of video frames. The simplest block matching algorithm (BMA) is also known as the full search (FS) algorithm [2]. Each $M \times N$ frame is subdivided into the square components $B(i, j)$ of size $(b \times b)$, $i = 1,\ldots, M/b$ and $j = 1,\ldots, N/b$. Using a frame of reference for every B_m block based on the current frame on the block distortion measurement (BDM), we can execute the search technique. The motion vector (MV) represents the motion from the present block to the next block in the most suitable frame of reference. Each pixel within a sequence of video frames is assigned an identical motion vector.

$$\forall \vec{r} \in B(i,j), \vec{d}(\vec{r}) = \vec{d}(i,j) \quad (1)$$

where pixel intensity is determined as $\vec{r} = (d_x, d_y)^T$, the time frame t is represented as $I(\vec{r}, t)$, and $\vec{d} = (d_x, d_y)^T$ with movement over time Δt.

Block number $(2w + 1)^2$ can be entered in the search box. Its size is $b \times b$ pixels, and its greatest motion vector movement is ±w pixels positioned vertically and horizontally. Fig. 2. shows the basic idea on which the block matching method is based [1, 9].

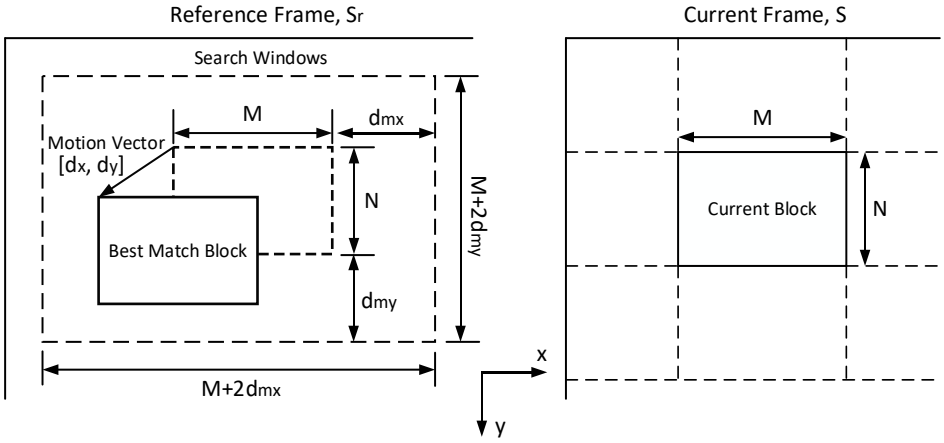


Figure. 2 Motion estimation using block matching

The cross-correlation, mean square error (MSE), and mean absolute error (MAE) are utilized as the corresponding factor parameters. To obtain good results on two block adjustments in the cross-correlation function (CCF) method, we multiply the correlation levels. In its implementation, we use MSE and MAE since CCF does not typically yield the best results in motion tracking, especially if the constant w is not high.

Eqs. (2) and (3) show that the function measures MAE and MSE, whereas the equation measures PSNR, where *N* is the frame's pixel number and 255 is its eight-bit resolution [4].

$$MSE(i,j) = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\left(f(m,n) - g(m+i,n+j)\right)^2, \quad -\mathrm{w} \le \mathrm{i,j} \le \mathrm{w} \tag{2}$$

$$MAE(i,j) = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}|f(m,n) - g(m+i,n+j)|, \quad -\mathrm{w} \le \mathrm{i,j} \le \mathrm{w} \tag{3}$$

$$PSNR = 10\log_{10}\left[\frac{255^2}{\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}(f(m,n)-g(m+i,n+j))^2}\right] \tag{4}$$

where $f(m, n)$ is the variable for the present block with an area of N^2 pixels located at positions (m, n) and $g(m + i, n + j)$ represents a block variable corresponding to the preceding video frame with new location points of $(m + i, n + j)$. The optimal matching procedure involves obtaining the positions $i = a$ and $j = b$, for instance, so that the movement vector denoted by MV(a, b) accurately represents the displacement of each pixel within a block.

To find the best match, we perform an exhaustive search with parameters $(2w + 1)^2$ where generally the price is $w = 7$ pixels [4]. The MAE measurement type is used to reduce the processing load, which is the standard video codec. Each N^2 block carries out the $(2w + 1)^2$ search, testing each procedure using $2N^2$ of addition and subtraction. The PSNR evaluation formula, where N represents the cumulative total of pixels in the video frame, provides the highest possible value of 255 and an eight-bit resolution.

2.3 Three-step search algorithm

The method in this paper utilizes a coarse-to-precise approximation based on logarithmic derivation. The TSS algorithm examines eight points surrounding the center of the video frame. For a middle position $[ax, ay]$ with a step d, $[ax – d, ay – d]$, $[ax – d, ay]$, $[ax – d, ay + d]$, $[ax, ay – d]$, $[ax, ay]$, $[ax, ay + d]$, $[ax + d, ay – d]$, $[ax + d, ay]$, $[ax + d, ay + d]$ are checked [1, 8]. After every stage, the size of the step is halved, and the stage with the least amount of warping is used as the center for the next stage. The process continues until the step size equals one. This decreases the number of search points to $[1 + 8\{\log_2(d + 1)\}]$.

The challenge with the TSS algorithm is that the first step requires the use of checkpoints that are all placed in the same way, which is not good for estimating small motion.

3. Proposed method

3.1 Inter-prediction video coding algorithm

For the estimation of and compensation for motion, the inter-predictive mode of the HEVC encoder uses previously reconstructed video frames as a reference.

The estimation of motion between two pictures is represented as $\psi(x, y, t_1)$ and $\psi(x, y, t_2)$.

The change in the direction vector at x between times t_1 and t_2 is equal to the change in the position of the point from t_1 to t_2. The video frame at time t_1 is referred to as the anchor frame, and the target frame is the video frame at t_2. If $t_1 < t_2$ in the video code, forward motion is considered. If $t_1 > t_2$, backward motion is considered [10], where the anchor frame is $\psi_1(x)$, the target frame is $\psi_2(x)$, the motion parameter is a motion vector on pixels in the anchor frame = $d(x)$, the field of motion is $d(x;a)$, $x \in A$, and the objective for configuring is $w(x;a) = x + d(x;a)$, $x \in A$.

The aim of any video encoding motion estimation approach is to reduce two parameters: SSD and sum-of-absolute-difference (SAD). Both parameters are computed within the present frame $I'(x',y')$ and the preceding frame $I(x)$ [10, 17, 18]:

$$SSD = \sum_{x,y \in R}\left(I(x,y) - I'(x',y')\right)^2 \quad (5)$$

$$SAD = \sum_{x,y \in R}\left(I(x,y) - I'(x',y')\right) \quad (6)$$

3.2 Inter-prediction video coding algorithm

The estimation of motion is a crucial system variable for transmission video coding. This study evaluates the motion estimation technique based on gradient descent and the implementation of the hierarchy for each video frame transmitted.

Our algorithm includes three stages. To enhance centering and decrease the complexity of computation, we first construct a three-level pyramidal structure of a low-pass image utilizing low-pass filters and downsampling. This strategy is referred to as three-level hierarchical implementation [10].

The initial translation is a whole pixel, with good accuracy at the pyramid's apex. The estimation uses a matching technique that lowers SAD by employing an altered n-step approach. At each pyramid level, the translation begins at the coarse level, working at a gradient point. To make the SSD easier to understand, the movement factors are found. Because it relies on the SSD, the motion parameter is not linear, and the iteration procedure is used as follows:

$$\boldsymbol{a^{(t+1)} = a^{(t)} + H^{(-1)} b} \quad (7)$$

where a is the motion parameter time t and time $t + 1$, H is a similar matrix $n \times n$ for the first half-time, b is the same n-element vector at minus half the first gradient time on SSD, and n is the number of parameters.

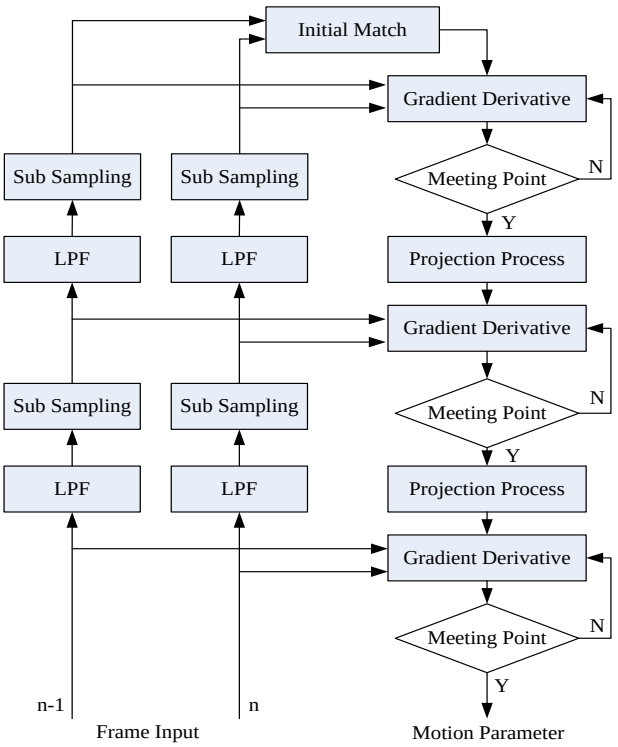


Figure. 3 Three-level hierarchy implementation flowchart

Fig. 3. shows how parameters a_0 and a_1 are set up initially using the interpretation of the predicted vector at the first place corresponding steps. The approach begins with the pyramid's base top level and descends through a sub-sequence of levels in the approach. The following steps show where the curve is at each level:

1. Calculate the H matrix and vector b.
2. Define the system by counting $H^{(-1)}b$.
3. Adjust a by adding parameters:

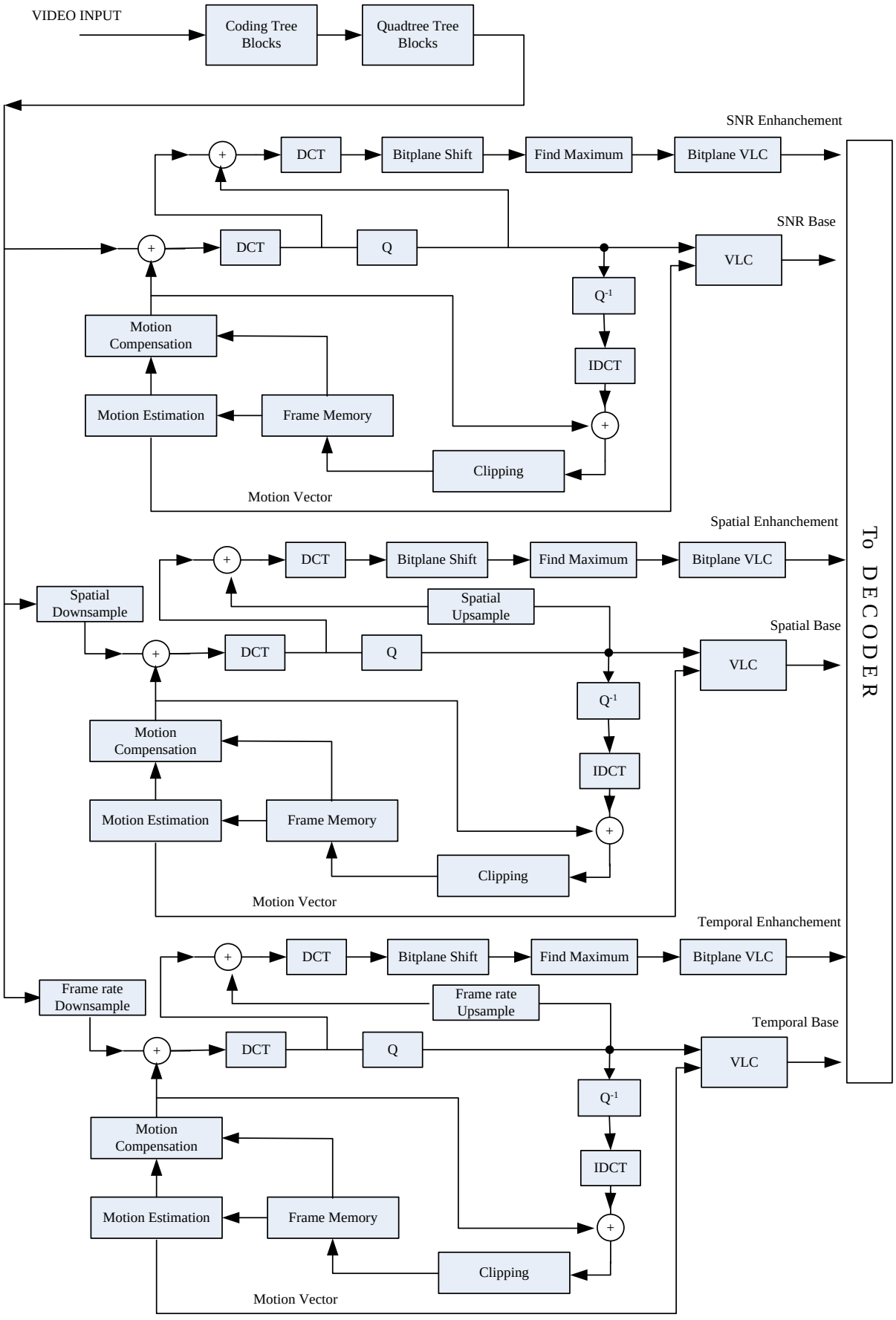
$$a^{(t+1)} = a^{(t)} + \delta a \quad (8)$$

where $\delta a = H^{(-1)}b$.

3.3 Multi-scalability design of video coding with inter-mode

We propose a new architecture in the video inter-coding mode with multi-scalability, as shown in Fig. 4. for the encoder and the decoder [6]. The output is modeled with multi-scalability, consisting of SNR, spatial, and temporal scalability, with each function having base and enhancement layers. The figure shows the modified part, namely the motion estimation component, in the box outlined in red, as proposed in this research. By changing the motion estimation algorithm and FS compensation, which is the video coding standard, using the block-matching

Received: May 14, 2024. Revised: June 14, 2024.



 DOI: 10.22266/ijies2024.0831.76

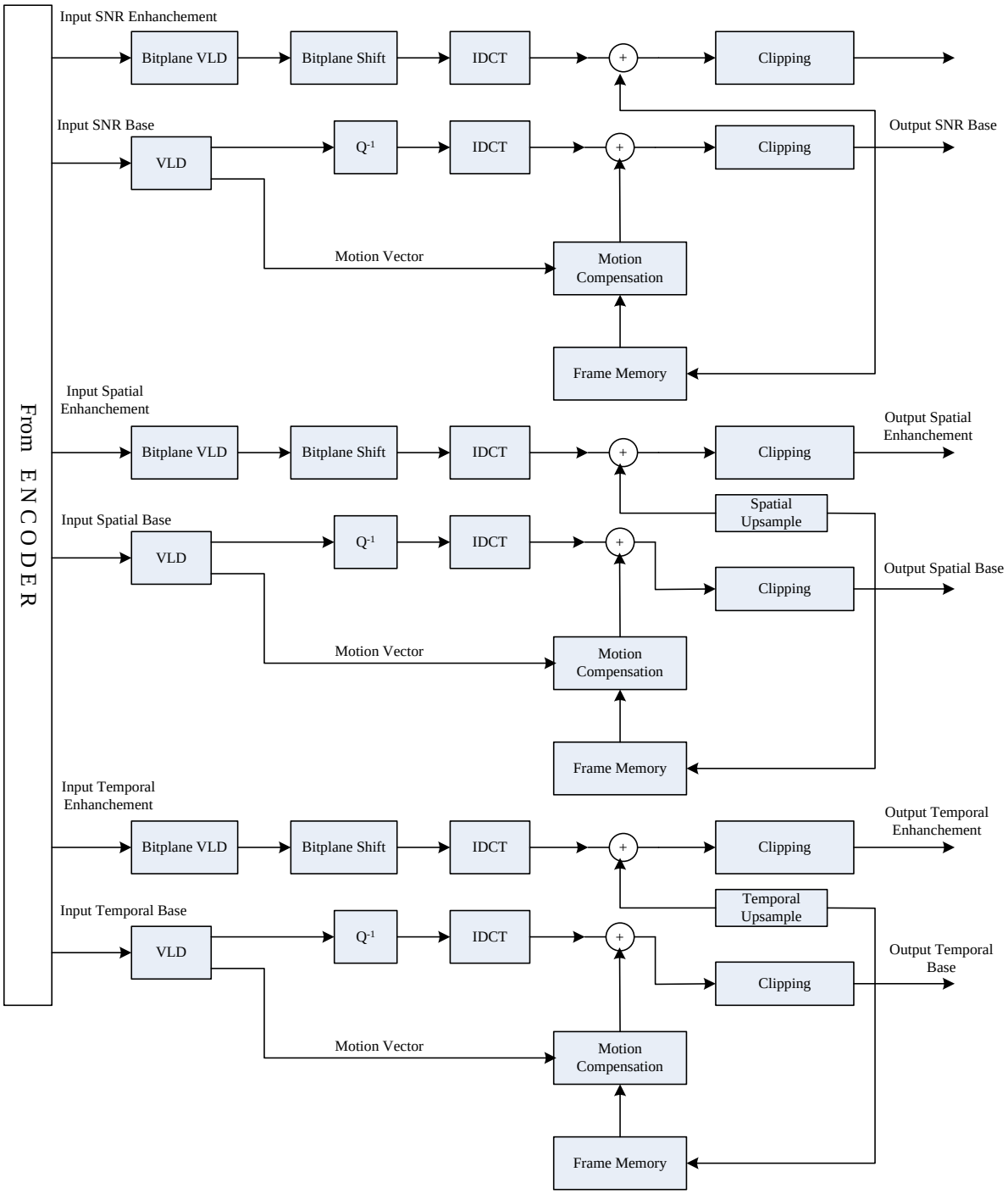


Figure. 4 Proposed HEVC multi-scalability encoder–decoder block diagram

TSS method, the system performance is obtained, yielding the speed efficiency and quality level of video coding. In the video coding analysis, we only use inter-frame mode, where the video frame coding is in the form of macroblocks, which are presented as some motion compensation predictions using motion vectors encoded in different frame representations, as shown in the following Eqs. (9) and (10) [19]:

$$E\{\hat{f}_n^i(b)\} = \hat{e}_n^i(b) + E\{\hat{f}_{n-1}^i(b)\} \quad (9)$$

$$E\{\hat{f}_n^i(e)\} = \hat{e}_n^i(b) + E\{\hat{f}_{n-1}^i(e)\} \quad (10)$$

where $\hat{e}_n^i$ is quantization and $\hat{f}_{n-1}^i(b)$ is the previous frame reference.

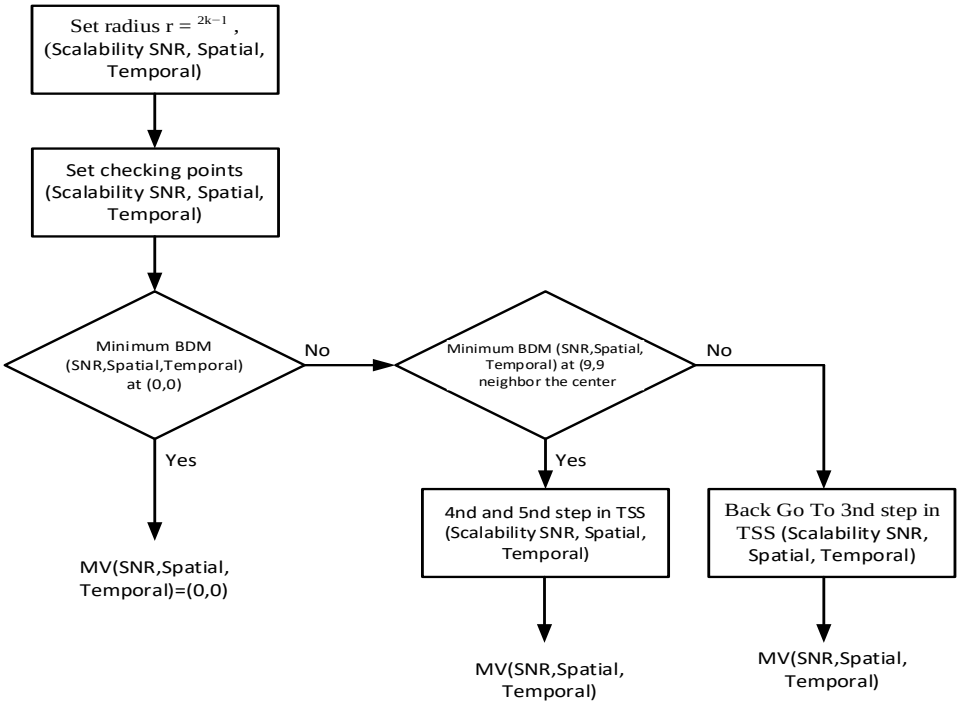


Figure. 5 TSS algorithm flowchart

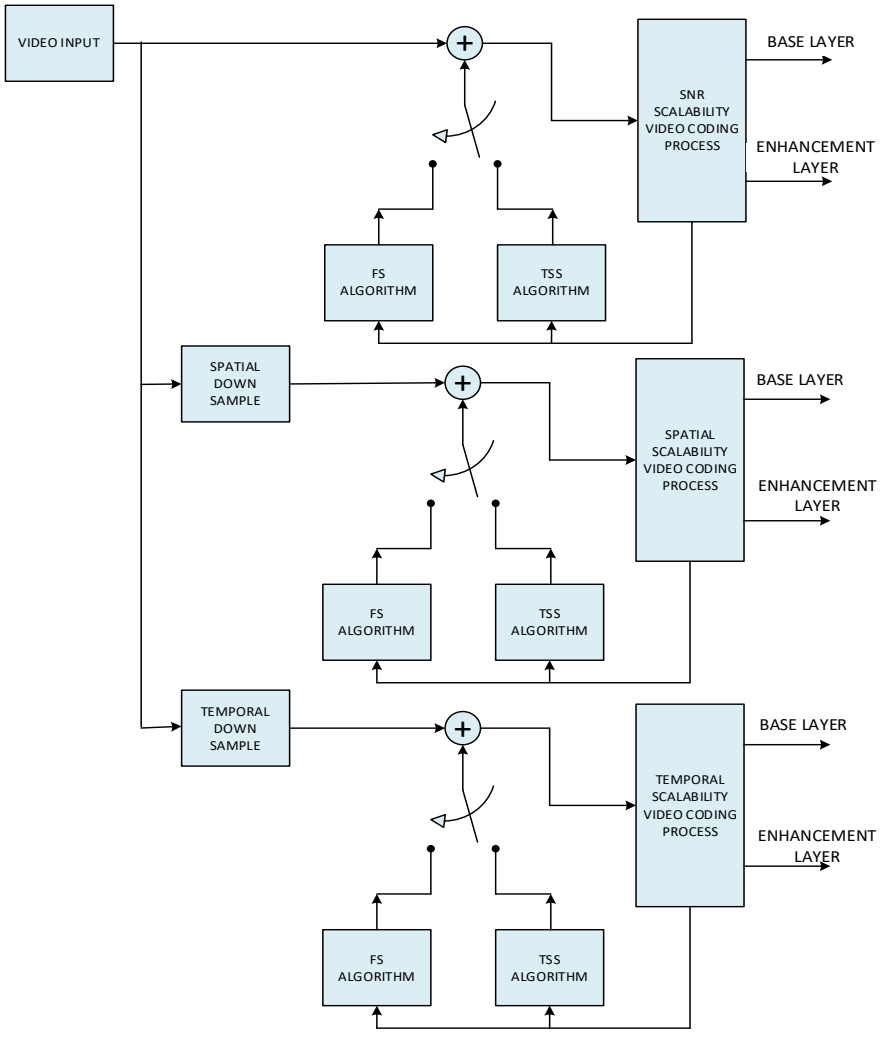


Figure. 6 Proposed HEVC multi-scalability encoder decoder–block diagram

In this algorithm, as shown in Fig. 5., we present a technique to estimate the motion of a series of frames efficiently. The method can be adapted to various movement models. This technique minimizes the SSD movement correction between the current and previous frames. Fig. 6. shows the modeling in a multi-scalability video coding experiment to observe the performance of two motion estimation algorithms, namely, the FS and TSS algorithms. Using these two algorithms alternately, we can evaluate performance in terms of efficiency and quality. Fig. 6. shows a series of block diagrams consisting of SNR scalability, which receives video input directly for processing. In addition, spatial scalability through a spatial downsampling process reduces the height and width of the video data for the base layer, and temporal scalability through a temporal downsampling process reduces the speed of the video input for the base layer.

3.4 Modeling the full search algorithm

The FS BMA eliminates the temporal redundancy found in standard video coding. Block matching is a process of dividing a video frame into blocks of the same size that do not overlap and calculating the adequate block shift from the previous structure block as a motion vector in the search frame's present frame. In the BM, each target block frame is compared with the previous frame to find the most suitable block. The matching criteria are based on the BDM, as explained in Eq. (12) and (13). The implementation of the FS algorithm can be explained in the following steps with SNR, spatial, and temporal scalability:

a. Configure the search area for $r = {}^{2k-1}$ where $k = (2d_m+1)^2$.
b. Setup checkpoints for $\Gamma = \{[0,0],[\pm r,\pm r],[0,\pm r],[\pm r,0]\}$.
c. Score the BDM with multiple scalabilities (SNR, spatial, and temporal) at each of the four candidate locations, choosing the lattice point with the lowest BDM and different scalabilities (SNR, spatial, temporal): $d' = arg\left(\min_{(i,j)\in\Gamma}(BDM(i,j))\right)$.
d. Adjust $r = \frac{r}{2}$.
e. If $r < 1$, then $d = d'$ and stop; else, adjust the search location:

$\Gamma = \{[0,0],[\pm \mathrm{r},\pm \mathrm{r}],[0,\pm \mathrm{r}],[\pm \mathrm{r},0]\}$, and return to Step c.

With each motion vector contained within the macroblock, $(2d_m+1)^2$ motion vector candidates are entered into the search box. The MN pixels are compared at each search location, and each comparison process requires three mathematical operations: substitution, addition, and taking the absolute value. Overall complexity in the process of basic arithmetic operations in each part of the macroblock is $3\mathrm{MN}(2d_m+1)^2$. Therefore, for a frame rate F (fps) and a frame size $I \times J$, the procedures per second in overall complexity can be described in the following Eq. (11):

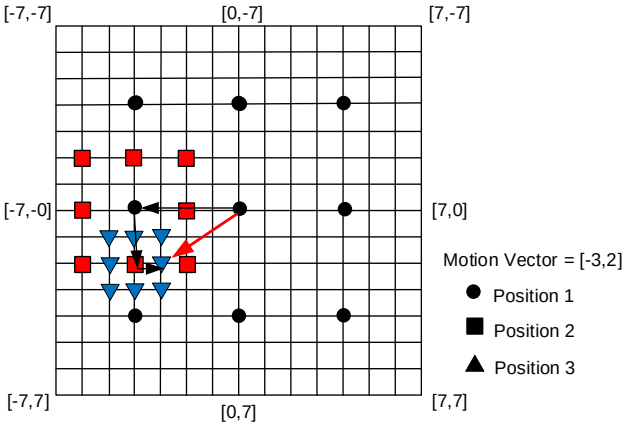


Figure. 7 TSS pattern

$$C_{Full\ Search} = 3IJF(2d_m+1)^2 \tag{11}$$

3.5 Modeling a three-step search algorithm

Block matching is proposed using a TSS algorithm. Fig. 7. shows that the search area reduces by a factor of two with each modification. When the search radius equals one, the TSS algorithm is complete. This technique has the advantage of requiring a fixed number of iterations for each operation, providing consistency in its software and hardware implementations, with the matching criteria based on the following BDM Eq. (12) [20]:

$$BDM(i,j) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} g(s(x,y) - s_r(x+i,y+j)) \tag{12}$$

which is a function that calculates BDM with g (.), where (i, j) represents the compensations for the candidate movement vector at that point, x and y represent the current block's local coordinates, and the motion vector is the variable determined to minimize the existing block's BDM:

$$d = [d_x, d_y]^T = arg\left(\min_{\forall(i,j)}(BDM(i,j))\right) \tag{13}$$

The method for implementing the TSS algorithm can be explained by the block diagram in Fig. 5.

The three-step search algorithm with SNR scalability, spatial scalability, and temporal scalability is as follows:

a. Configure the search area for $r = {}^{2k-1}$ where k = $[\log_2(d_m)]$.
b. Setup checkpoints for
$\Gamma = \{[0,0],[\pm r,\pm r],[0,\pm r],[\pm r,0]\}$.
c. Score the BDM with multiple scalabilities (SNR, spatial, and temporal) at each of the nine candidate locations, choosing the lattice point with the lowest BDM and different scalabilities (SNR, spatial, temporal): $d' = arg\left(\min_{(i,j)\in\Gamma}(BDM(i,j))\right)$.
d. Adjust $r = \frac{r}{2}$.
e. If $r < 1$, then $d = d'$ and stop; else, adjust the search location: $\Gamma = \{[0,0],[\pm r,\pm r],[0,\pm r],[\pm r,0]\}$, and return to Step c.

The TSS method complexity can be calculated using the following Eq. (14):

$$C_{Three\ Step\ Search} = 3IJF(8k+1) \quad (14)$$

where *C* is the system's complexity, *I* is the *m*th matrix, *J* is the *n*th matrix, *F* is the frame rate (fps), and *k* is the number of steps used. Fig. 7. illustrates a TSS model, showing how the search algorithm converges on the motion vector *d* = [3,2], assuming a 15 × 15 search area and a decreasing error rate. In *N* stages, the TSS algorithm converges for each video frame. However, the convergence profile differs depending on the specific geometry and error rate.

3.6 Bjontegaard delta rate calculation

In the suggested simulation for H.26L', one way to improve performance is to create an RD-plot that shows the difference in PSNR and bit rate between the two simulation conditions to determine how much the two shapes vary from each other on average [21, 22]. The fundamental aspects of the analysis are as follows:

- We adjust the curve using four data points (the PSNR or bit rate is the value for the quantization parameter equal to 8, 16, 32, and 64).
- The integral expression of the curve is known based on the PSNR and bit rate values.
- The average disparity is the difference between the integrated periods and the integration results.

With a linear bit-rate scale, third-order polynomials in the following form can be used to make straight-line interpolations:

$$SNR = a_1 + a_2 \text{ bit} + a_3 \text{ bit2} + a_4 \text{ bit3} \quad (15)$$

where a_1, a_2, a_3, *and* a_4 are curves passing through the four datapoints.

As a result, the following values can be obtained:

- Bjontegaard delta (BD)-PSNR, which is the average PSNR difference in decibels across the entire bit-rate range.

BD-rate, the average bandwidth difference across the entire PSNR range.

4. Experiment results

4.1 Experimental data and setup

This section evaluates multi-scalable HEVC inter-mode with motion estimation block modeling using the FS and TSS algorithms with the proposed method, comparing the two motion estimation algorithms. Tests were carried out on the PSNR and bit-rate performance sections for the two motion estimation algorithm models, TSS and FS algorithms, which are proposed to adapt to different video content. The obtained results show that the proposed technique increases HEVC quality and effectiveness with multi-scalability using a motion estimation algorithm.

The video test for the simulation used in the experiment is shown in Fig. 8. We conducted simulations using the HM software reference library on the encoder and decoder for the analysis with MATLAB and Excel software with video test media (https://media.xiph.org/) [23, 24], where the *Y* component (luminance) at 100 frames was at 30 fps for SNR and spatial scalabilities and 15 fps for temporal scalability.

4.2 Performance evaluation

Fig. 8. shows the simulation results of the estimation and motion compensation experiments on video coding using the gradient descent method for the video sequence test on the *Y* component (luminance), obtaining the reconstruction of the target frame, anchor frame, motion vector, and predictive frames.

Fig. 9. show the coding of the I and P frames with the CTU structure and block size distribution in quadtree coding in the Akiyo video sequence. Meanwhile, Fig. 10. show that the number of motion vector directions of the FS and TSS algorithms in the video test series in the Akiyo sequence affects the PSNR. The simulation results of the motion estimation and motion compensation in frames 3 to 4

Akiyo, 352×288 (CIF-SD), 300 frames

Bus, 352×288 (CIF-SD), 150 frames

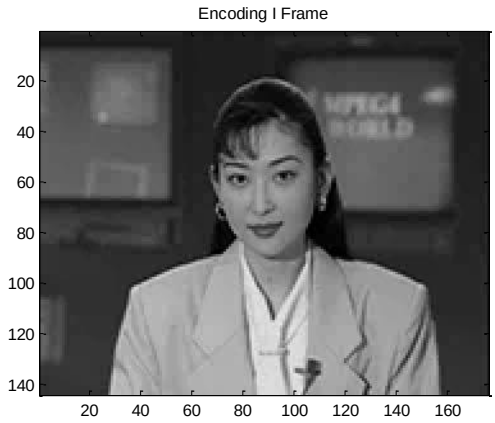
Flower Garden, 720×432 (4CIF-SD), 360 frames

Football, 720×432 (4CIF-SD), 360 frames

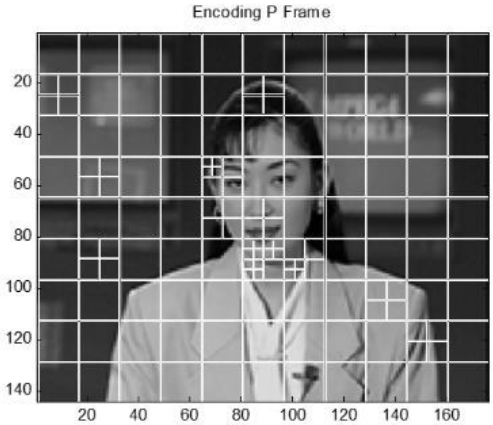
Shields, 1280×720 (HD), 504 frames

Four People, 1280×720 (HD), 600 frames

Figure. 8 Test videos for simulation systems

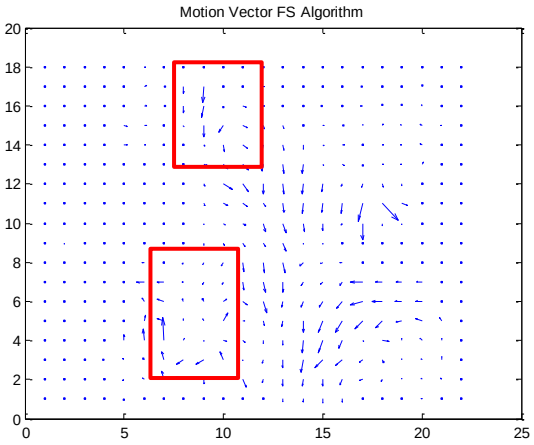


Encoding I–P frame

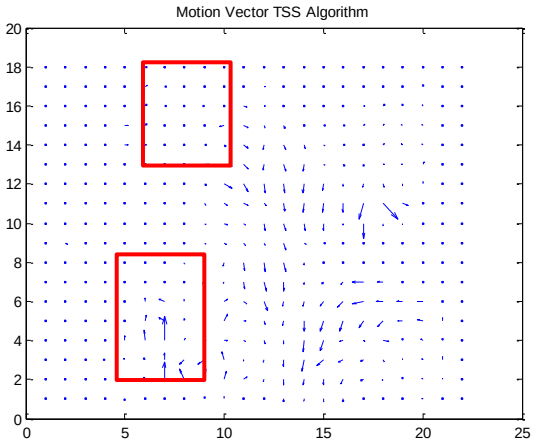


Block size distribution

Figure. 9 Display encoding



FS algorithm with a time interval of 0.796 s



TSS algorithm with a time interval of 1.444 s

Figure. 10 Motion vector field “sequence Akiyo” frames 3 to 4 at 30 fps

Table 2. Full search experimental results

Video Test	Multi-scalability	Average PSNR (dB)		Average MSE		Average Bit Rate (kbps)		CPU Time (s)		BD-PSNR (dB)	BD-rate (%)
		Base	Enhancement	Base	Enhancement	Base	Enhancement	Base	Enhancement		
Akiyo	SNR	36.6	73.1	0.9	0.8	268.	537.	1827.	1771.	0.3	78.8
	Spatial	36.6	34.1	0.9	0.9	268.	6418.	3550.	2958.	−2.2	37.8
	Temporal	18.5	36.5	1	0.9	157.	268.	837.	1739.	0.2	53.3
Bus	SNR	32.5	64.9	0.9	0.9	637.	11,274.	2921.	1540.	0.2	−1
	Spatial	32.5	32.2	0.9	0.9	5637.	7618.	1501.	3068.	−0.1	0.5
	Temporal	16.2	32.4	1	0.9	2,836.	5636.	746.	1513.	0.1	−1
Football	SNR	32.8	65.4	0.9	0.9	21,039.	42,081.	381.	7167.	0.2	−1
	Spatial	32.8	33.7	0.9	0.9	21,039.	6087.	9908.	2486.	0.1	−0.7
	Temporal	16.4	32.6	1	0.9	10,236.	21,042.	3514.	7174.	0.1	−1
Flower Garden	SNR	33.1	66.4	0.9	0.9	28,319.	56,641.	14,735.	16,425.	0.1	−0.1
	Spatial	33.1	34.3	0.9	0.9	28,319.	8857.	14,729.	4595.	0.2	−0.7
	Temporal	16.6	33.3	1	0.9	13,968.	28,321.	7182.	16,661.	−0.005	1.1
Shields	SNR	27.8	60.4	0.9	0.9	29,134.	66,777.	14,972.	14,880.	0.2	−1
	Spatial	32.6	34.3	0.9	0.9	29,134.	22,795.	14,880.	11,048.	0.1	−0.4
	Temporal	16.3	32.6	1	0.9	23,555.	37,640.	8447.	17,005.	0.2	−0.6
Four People	SNR	34.7	69.2	0.9	0.9	11,45	22,896.	12,732.	12,429.	0.3	−1
	Spatial	34.7	36,5	0.9	0.9	11,45	19,816.	16,483.	10,893.	−0.04	0.6
	Temporal	17.6	34.6	1	0.9	582	11,445.	6150.	12,700.	0.2	−1

Table 3. Three-step search experimental results

Video Test	Multi-scalability	Average PSNR (dB)		Average MSE		Average Bit Rate (kbps)		CPU Time (s)		BD-PSNR (dB)	BD-rate (%)
		Base	Enhancement	Base	Enhancement	Base	Enhancement	Base	Enhancement		
Akiyo	SNR	34	68	0.9	0.9	203.	406.	1743.	179	0.2	50.7
	Spatial	34	30.4	0.9	0.9	203.	5587.	1847.	3045.	0.4	26.3
	Temporal	17.4	34	1	0.9	126.	203.	835.	1676.	0.2	4.3
Bus	SNR	27.3	55.2	0.9	0.9	4653.	9307.	1591.	162	0.1	−1
	Spatial	27.3	28.7	0.9	0.9	4653.	6392.	1549.	305	−0.1	0.6
	Temporal	13.8	27.8	1	0.9	2353.	4653.	833.	1550.	0.1	−1
Football	SNR	28.5	57.3	0.9	0.9	17,376.	34,754.	7434.	7722.	0.2	−1
	Spatial	28.5	30.6	0.9	0.9	17,376.	5356.0	7422.	2540.	0.1	−0.8
	Temporal	14.3	28.8	1	0.9	8487.	17,377.	3707.	733	0.1	−1
Flower Garden	SNR	29.6	59.5	0.9	0.9	24,428.	48,858,	14,454.	14,601.	0.2	−1
	Spatial	29.6	31.1	0.9	0.9	24,428.	7767.	14,384.	4434.	0.1	−0.7
	Temporal	14.9	29.9	1	0.9	12,089.	24,430.	7141.	15,801.	0.1	−1
Shields	SNR	28.3	56.2	0.9	0.9	30,075.	60,142,	16,93	13,804.	0.2	−1
	Spatial	28.3	30.8	0.9	0.9	38,55	19,557.	15,451.	11,055.	0.05	−0.5
	Temporal	14.3	27.9	1	0.9	25,419.	30,067.	6966.	13,766.	0.12	−1
Four People	SNR	30.1	60.8	0.9	0.9	9696.	19,388.	12,496.	15,957.	0.2	−1
	Spatial	30.1	32.2	0.9	0.9	9696.	16,53	12,538.	10,083.	−0.03	0.3
	Temporal	15.2	30.7	1	0.9	4966.	969	6173.	13,147.	0.1	−1

of the Akiyo sequence show that the FS algorithm is more accurate than the TSS algorithm, as indicated by the presence of a large motion vector (red squares in Fig. 10.). Motion vectors were recorded at time intervals of 0.796 s for FS and 1.444 s for TSS at 30 fps.

The system simulation results show the HEVC performance with multi-scalability using FS and TSS algorithms on motion estimation. Simulation experiments were conducted to obtain objective video frame quality values by averaging the parameters of PSNR, MSE, CPU time, BD-PSNR,

and BD rate in every scalability and layer, as shown in Tables 2 and 3.

Table 2 presents the results of motion estimation using the FS algorithm with multi-scalability video coding. The average PSNR for all base layers ranged from 16.6 to 36.6 dB, and all enhancement layers ranged from 32.2. to 73.1 dB. The average MSE on all scalabilities with the base layer maximized at a value of 1 with a minimum of 0.9, and the enhancement layer had a maximum of 0.9 and a minimum of 0.8. The average bit rate for all base layers ranged from 157.6 to 29,134.5 kbps, and for all enhancement layers, the maximum value was 66,777.3 kbps, and the minimum was 268.7 kbps.

The average CPU time on all scalabilities with the base layer ranged from 746.3 to 16,483.8 s, and the enhancement layer had a maximum value of 17,005.5 s and a minimum of 1513.5 s. While the highest BD-PSNR was 0.3 dB and the lowest value was −0.005 dB, the BD rate ranged from −0.7% to 78.8%.

Table 3 shows the experimental results of the TSS algorithm with multi-scalable video coding. The average PSNR for all base layers ranged from 13.8 to 34 dB, and for all enhancement layers, the range was 27.8 to 68 dB. The average MSE on all scalabilities with the base layer was 0.9–1, and the enhancement layer's MSE was 0.9 for each case.

The average bit rate for all base layers was 126.6 to 30,075.1 kbps, and for all enhancement layers, it was 268.7 to 60,142.4 kbps. The average CPU time on all scalabilities with the base layer ranged from 835.6 to 14,384.1 s, and the enhancement layer's CPU time was 1628 to 15,957.8 s. At the same time, the BD-PSNR was −0.3 to 0.4 dB, with a BD rate of −0.8% to 50.7%.

The average amount is represented using the table of the FS algorithm's bit rate on multi-scalability consisting of SNR, spatial, and temporal at the base layer, which has a maximum value of 30.6 dB and a minimum of 17.6 dB. In comparison, the enhancement layer's maximum value is 47.9 dB and a minimum of 35.6 dB. For the average bit rate value of the TSS algorithm on multi-scalability consisting of SNR, spatial, and temporal at the base layer, the maximum value is 28.5 dB, and the minimum value is 22.8 dB.

In comparison, the enhancement layer's maximum value is 44.1 dB, and the minimum value is 37.2 dB. From the results of these measurements, the average bit rate for the enhancement layer in the FS algorithm and TSS algorithm has a high bit rate value.

The simulation results of HEVC system performance with multi-scalability using the FS method, namely the PSNR and MSE parameters, are shown in Fig. 11. and 12. The figures show that the SNR scalability achieved the highest PSNR and the lowest error rate in the enhancement layer compared to other scalability layers. The highest PSNR occurred in the Akiyo series, and the lowest was in the Shields series. This is because the Shields sequence displays more variations in image textures and, thus, more motion vectors than other sequences used as test videos for the FS algorithm. The graphs also show that the SNR and spatial scalability of the base layer have the same value. In addition, we see that a video sequence with a high PSNR exhibits a low MSE, and a video sequence with a high MSE has a low PSNR value.

Fig. 13. and 14 illustrate the simulation outcomes for the HEVC system with multi-scalability for the PSNR and MSE parameters using the TSS algorithm. The chart shows that the SNR scalability achieved the highest PSNR and the lowest error rate in the enhancement layer. The chart shows that the SNR scalability achieved the highest PSNR and the lowest error rate in the enhancement layer

Fig. 13. and 14 illustrate the simulation outcomes for the HEVC system with multi-scalability for the PSNR and MSE parameters using the TSS algorithm. The chart shows that the SNR scalability achieved the highest PSNR and the lowest error rate in the enhancement layer. The Akiyo sequence exhibits the highest PSNR than the other arrangements. This is because the Akiyo sequence has fewer image textures and fewer motion vectors compared to the different lines. Meanwhile, the lowest PSNR is in the Shields sequence, which has many image textures and significantly more motion vectors compared to other video sequences. The graphs also show that the PSNR values for the spatial scalability of the base layer, the SNR scalability of the base layer, and the temporal scalability of the additional layer have the same value. This is because scalability and layers with the same weight have the same coding structure, as shown in Fig. 5., where each scalability layer has two layers—a primary layer and another layer that is a modification of the basic model to create a system that has scalability with two different outputs. The graphs show that a video test series with a high PSNR has a low MSE, and a video test series with a high MSE value has a low PSNR.

Meanwhile, Fig. 16. displays the simulation results of HEVC with multi-scalability PSNR parameters in the FS and TSS algorithms, showing the average PSNR and bit rate for multi-scalability with the FS and TSS algorithms at all scalabilities, i.e., SNR, spatial, and temporal scalabilities, for all base and enhancement layers.

Fig. 15. shows that the FS algorithm in the enhancement layer for all scalabilities has high PSNR and bit rate values. The next highest PSNR and bit rate values are achieved by the TSS algorithm in the enhancement layer for scalabilities, followed by the base layer. The lowest PSNRs and bit rates are from the algorithm proposed by TSS in the enhancement layer for all SNR, spatial, and temporal scalabilities.

Table 4 compares the results of the BMA and FS and TSS algorithms on the HEVC system with multiple scalabilities. In the experiments, the FS and TSS algorithms were carried out to obtain the average bit rate and PSNR, average machine speed for accessing multi-scalability coding simulations, BD-rate, and BD-PSNR in video coding. The highest average PSNR of 47.9 dB was found in the Akiyo sequence with the FS algorithm in the enhancement layer, and the lowest of 17.6 dB occurred in the Four People sequence with the FS algorithm in the base layer. The Shields sequence shows the highest average bit rate with the FS algorithm at the base layer at 42.4 Mbps, and the lowest is from the Akiyo sequence for the TSS algorithm at the base layer at 177.6 kbps. For the average machine process for simulation, the longest time required was for the Shields sequence of the FS algorithm at the base layer at 14,000 s, and the lowest was the Bus sequence for the TSS algorithm at the base layer at 1475 s. The highest BD-PSNR was seen in the Akiyo sequence for the TSS algorithm at 0.3 dB, and the lowest was for the Akiyo sequence for the FS algorithm at −0.6 dB.

The simulation result shown in Table 4 are summary of the motion estimation algorithm applied to multi-scalability HEVC video coding, which is the method proposed in this research. This method is the development of a motion estimation algorithm system in single scalable video coding, which has been studied by previous researchers [4, 25-28].

The highest value BD-rate was with the Akiyo sequence in the FS algorithm at 64.9 dB, and the lowest was with the Football sequence in the FS and TSS algorithms at −0.9 dB. The enhancement layer in the FS algorithm has the highest average bit rate and PSNR compared to all layers in both algorithms. This difference is because the FS algorithm has high accuracy in reading the motion vector on each video frame compared to the TSS algorithm, which takes around three steps to read the motion vector on each video frame.

Fig. 16. shows a comparison of the FS algorithm and the proposed block-matching TSS algorithm for video coding with multi-scalability. To evaluate the system, the chart shows four evaluation parameters for each video test sequence: (i) the average corresponding result of the FS algorithm on the base layer with multi-scalability, (ii) the average of the FS algorithm on the enhancement layer with multi-scalability, (iii) the average of the proposed TSS algorithm on the base layer with multi scalability, and (iv) the average of the algorithm. The proposed TSS layer enhancement with multi-scalability for everything is applied to the evaluation of PSNR, bit rate, and CPU time. In increasing the PSNR layer, the FS and TSS algorithms have the highest PSNRs, which are almost evenly distributed for all video tests in the Akiyo, Bus, Football, Flower Garden, Shields, and Four People sequences. The FS and TSS algorithms also have the highest bit rates for video testing in the Shields sequence and the lowest values in the Akiyo video sequence for the performance parameters of the bit rate enhancement layer.

Table 4. Summary of the motion estimation algorithm

Video Test	Algorithm	Average PSNR (dB)		Average Bit Rate (kbps)		Average CPU Time (s)		BD-PSNR (dB)	BD-rate (%)
		Base	Enhancement	Base	Enhancement	Base	Enhancement		
Akiyo	FS	30.6	47.9	231.7	2408.3	2071.6	2156.4	−0.6	64.9
	TSS	28.5	44.1	177.6	2065.8	1475.6	2173.1	0.3	27.1
Bus	FS	27.1	43.2	4704.2	8176.7	1722.9	2040.6	0.1	−0.5
	TSS	22.8	37.2	3887.0	6784.6	1324.8	2076.9	0.1	−0.4
Football	FS	27.3	43.9	17,438.7	23,070.5	6934.9	5609.2	0.2	−0.9
	TSS	23.8	38.9	14,413.4	19,162.7	6188.1	5866.7	0.1	−0.9
Flower Garden	FS	27.6	44.7	23,535.8	31,273.3	12,216.0	12,561.0	0.1	0.1
	TSS	24.7	40.2	20,315.2	27,018.9	11,993.2	11,612.7	0.1	−0.9
Shields	FS	30.2	47.4	27,274.8	42,404.6	12,766.9	14,311.5	0.1	−0.7
	TSS	23.6	38.3	31,350.2	36,589.0	13,116.3	12,875.6	0.1	−0.8
Four People	FS	17.6	35.6	9574.0	18,052.8	11,788.8	12,008.0	0.1	−0.5
	TSS	25.1	41.2	8119.5	15,205.4	10,402.9	13,062.9	0.1	−0.5

Figure. 11 Performance FS multi-scalability PSNR graphs on Akiyo, Bus, Football, Flower Garden, Shields and Four People sequence

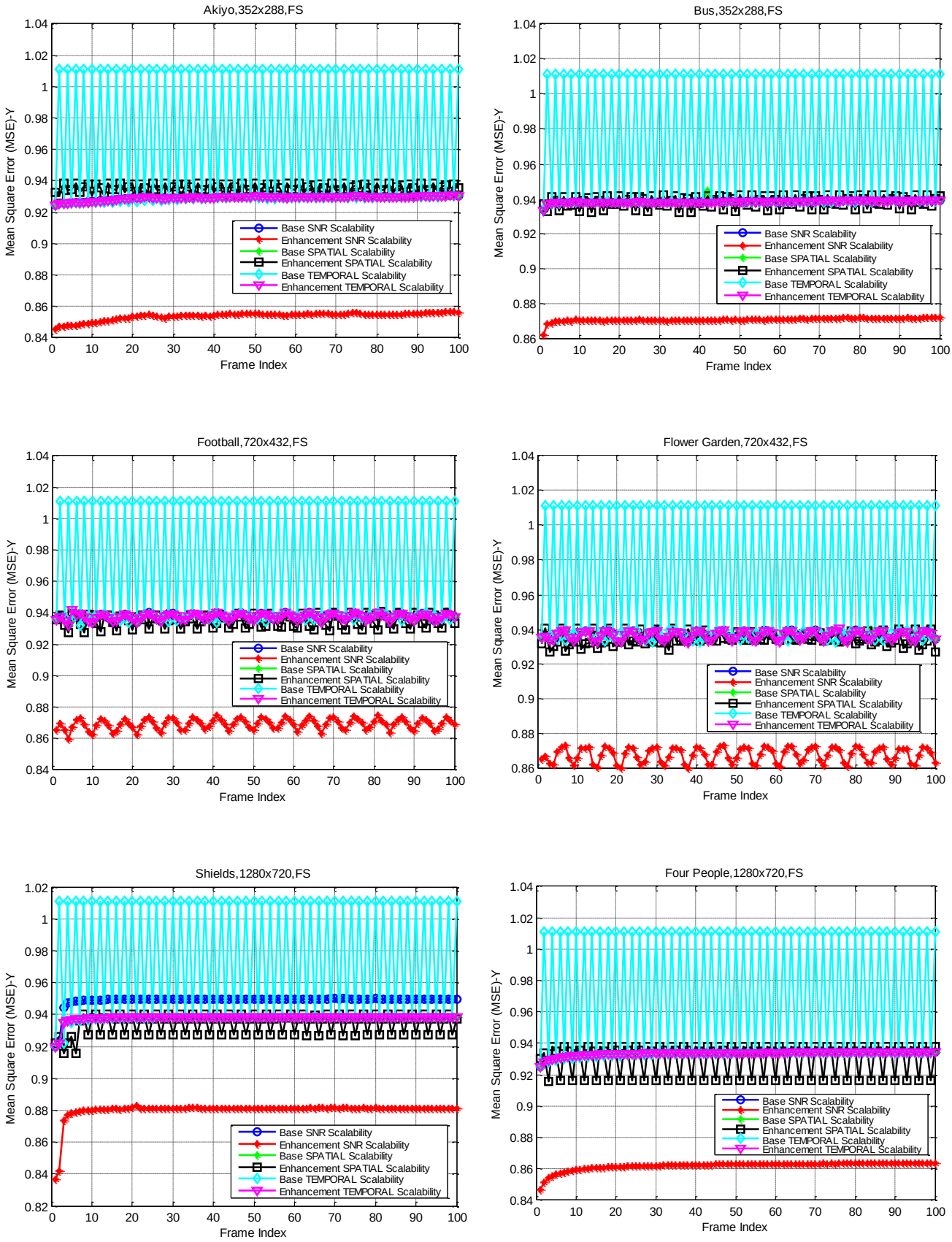


Figure. 12 Performance FS multi-scalability MSE graph on Akiyo, Bus, Football, Flower Garden, Shields and Four People sequence

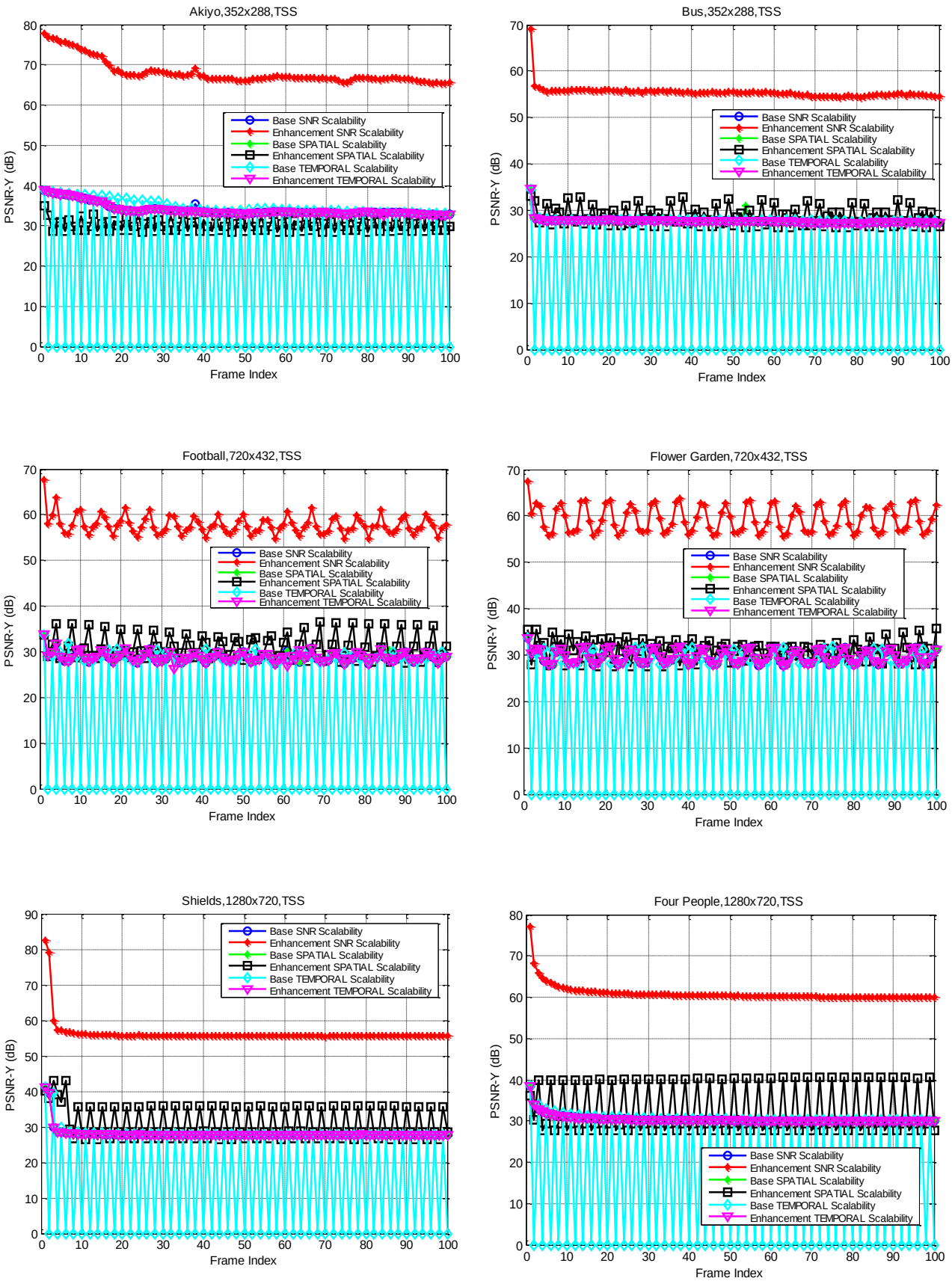


Figure. 13 Performance TSS multi-scalability PSNR graph on Akiyo, Bus, Football, Flower Garden, Shields and Four People sequence

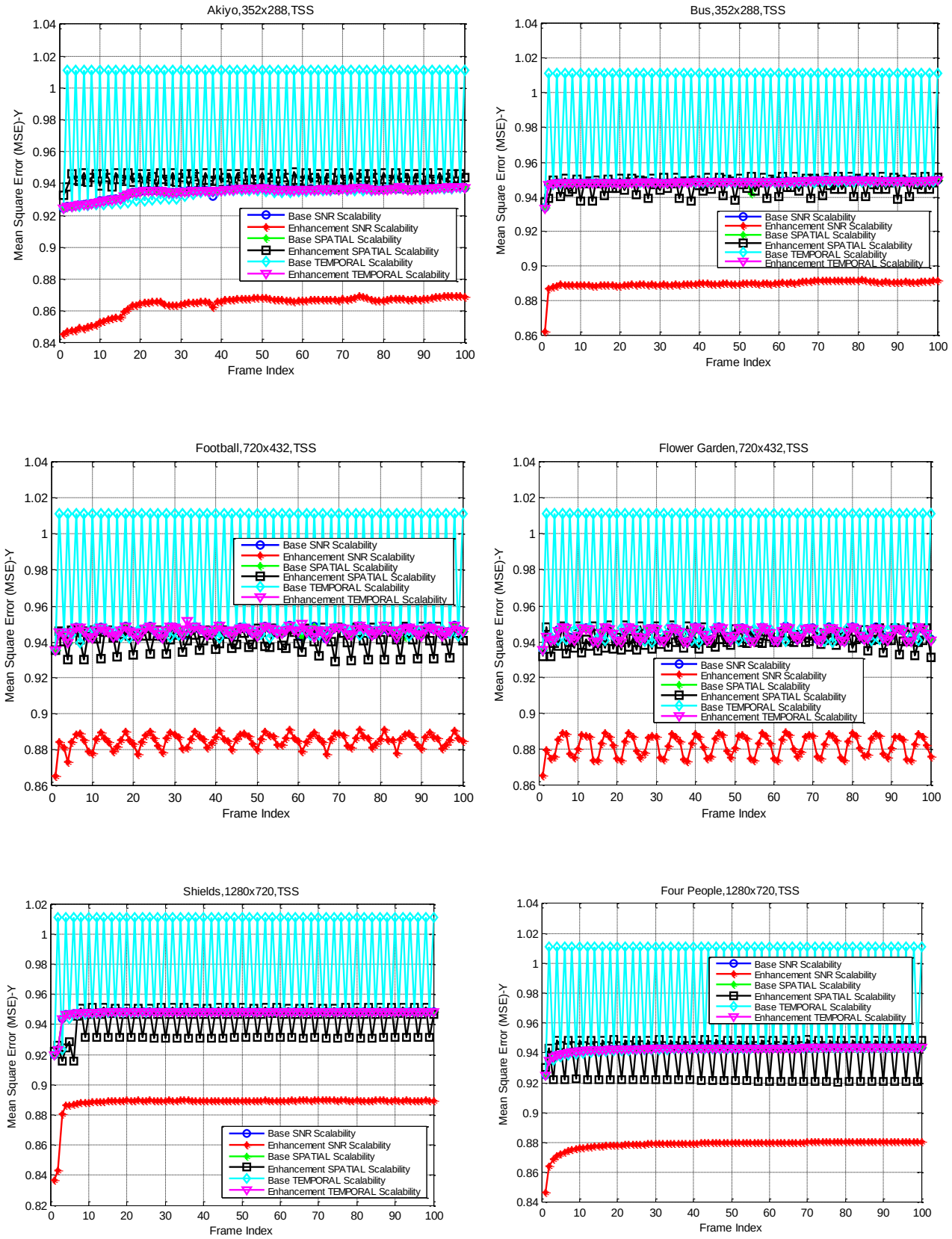


Figure. 14 Performance TSS multi-scalability MSE graph on Akiyo, Bus, Football, Flower Garden, Shields and Four People sequence

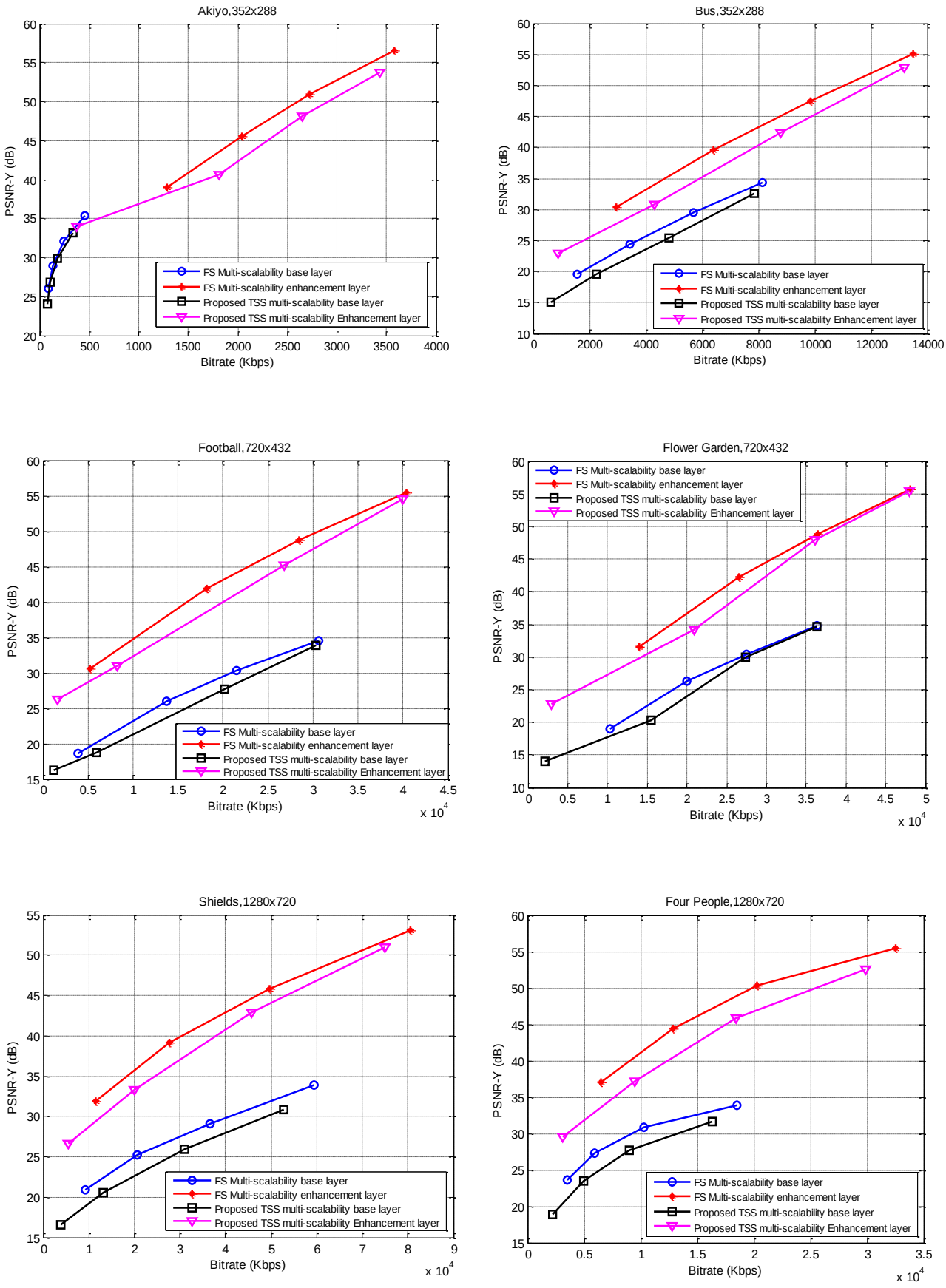


Figure. 15 Multi-scalability graph of bit rate performance differences in the FS method and the proposed TSS method on Akiyo, Bus, Football, Flower Garden, Shields and Four People sequence

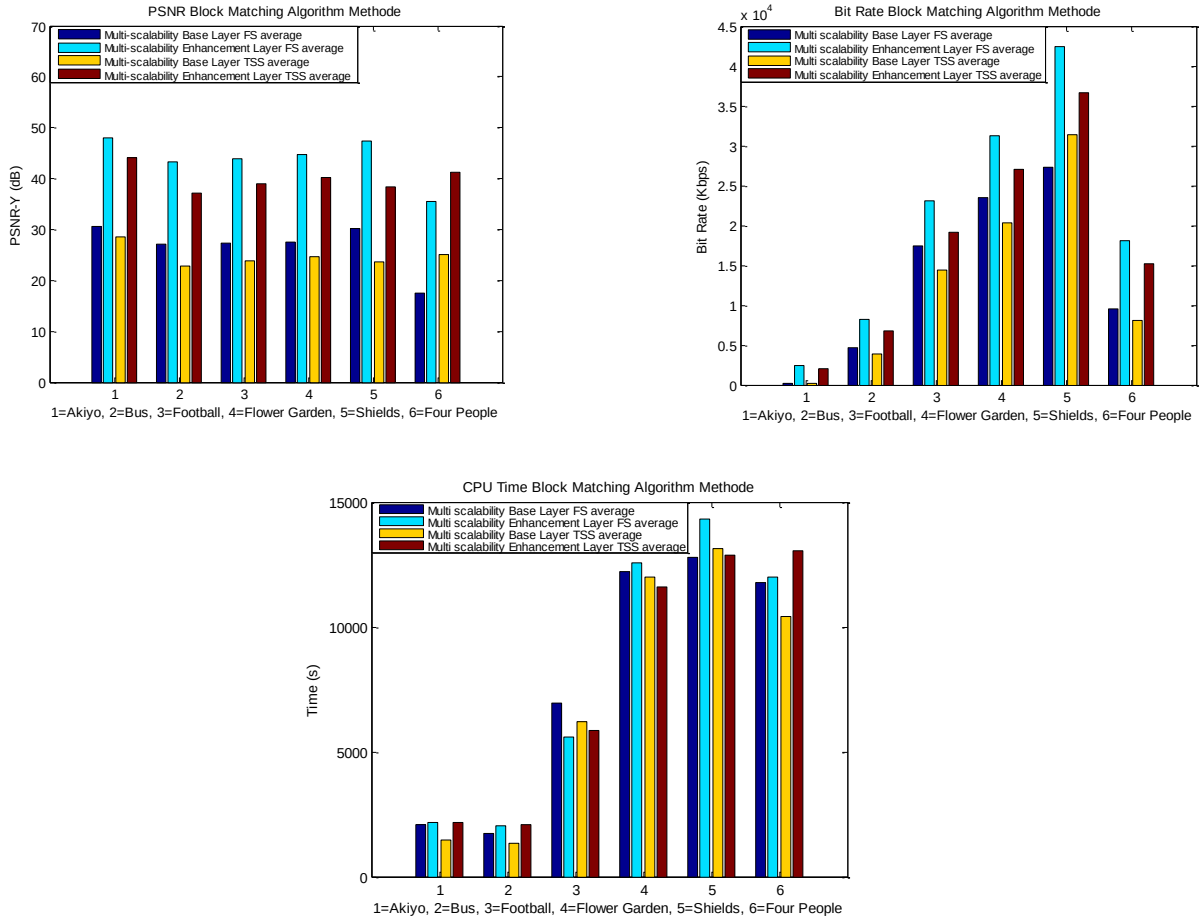


Figure. 16 Comparison of Block Matching FS and TSS Multi Scalability Algorithms in PSNR, Bit Rate and CPU Time parameters

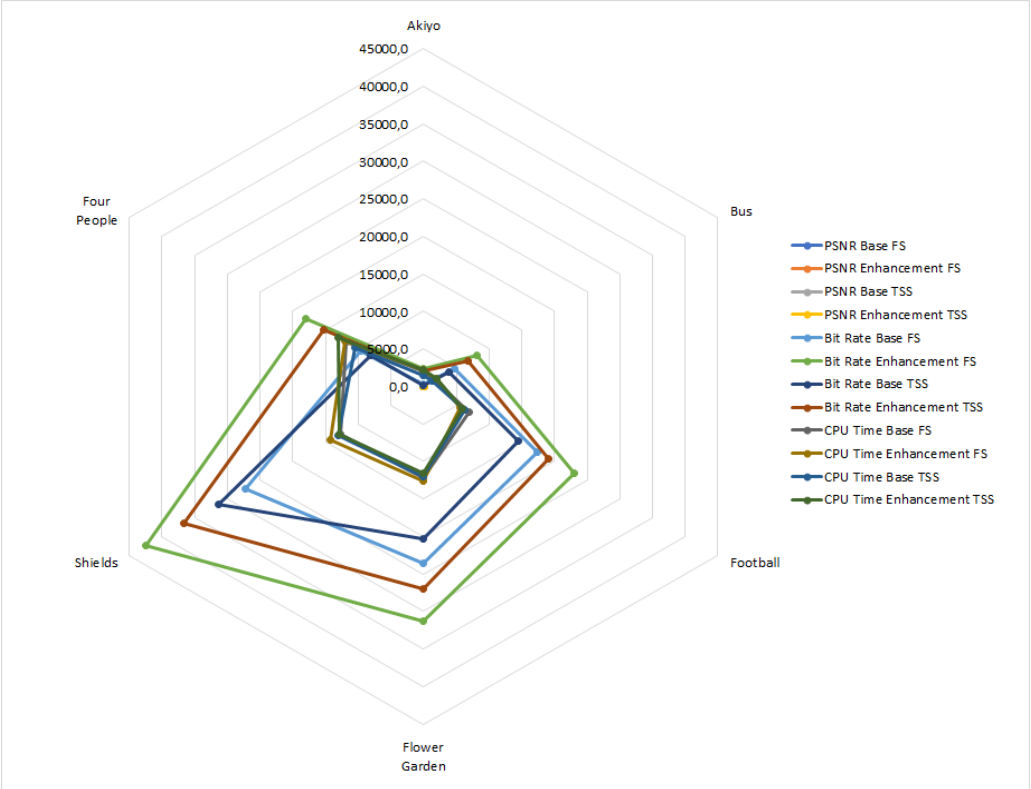


Figure. 17 FS and TSS performance radar chart patterns

For performance, the time used by the computer to process the simulation system using the FS algorithm and the proposed TSS algorithm was the longest for the Shields sequence. In contrast, the Bus sequence had the shortest time.

Fig. 17. shows a radar chart patterns comparison of the FS and TSS BMAs for video coding with multiple scalabilities with measurement parameters of PSNR, bit rate, and CPU time. The FS enhancement layer shows the highest bit rate, CPU time, and PSNR in all video tests used compared to layers and algorithms on other multi-scalabilities. Each video test had a different number of frames and motion vectors, so in the video encoding process with the FS and TSS methods, the transmission produced other performance results on the video tests of Akiyo, Bus, Football, Flower Garden, Shields, and Four People.

5. Conclusion

The experimental results show that the use of the FS algorithm still exhibits better performance than the proposed TSS algorithm, showing that the enhancement layer has a higher average bit rate and average PSNR but also a higher CPU time. In the system evaluation, the highest average bit rate and PSNR values after layer enhancement in the FS algorithm occur with the enhancement layer in the TSS algorithm. The FS algorithm has a total BD-PSNR of 0 dB and an efficiency of 62.4%, while the TSS has a total BD-PSNR of 0.8 dB and an efficiency of 23.6%. Therefore, in video transmission with multi-scalability coding, the FS algorithm performs better than the proposed TSS algorithm, and designing a system with strict performance in HEVC technology with multi-scalability requires layer enhancement with the FS and TSS algorithms.

Notation list

No	Notation	Meaning
1	$M \times N$	Original block size video frame
2	$\forall r^{\rightarrow}$	Motion vector
3	$(2w+1)^2$	Block number
4	$MSE(i,j)$	Mean Square Error (i,j)
5	$MAE(i,j)$	Mean Absolute Error (i,j)
6	$MV(a, b)$	Movement Vector (a, b)
7	$BDM(i,j)$	Block Distortion Measure (i,j)
8	$[ax, ay]$	Middle position pixel
9	$\psi(x, y, t_1)$, $\psi(x, y, t_2)$	Motion between two pictures
10	$I'(x',y')$	Present frame
11	$I(x)$	Preceding frame
12	$a^{(t+1)}$	Iteration motion estimation
13	$E\{\hat{f}_n^i(b)\}$	Previous frame reference motion vectors encoded
14	$E\{\hat{f}_n^i(e)\}$	Quantization motion vectors encoded
15	$r = 2^{k-1}$	Configure the search area
16	Γ	Pixel Setup checkpoints
17	d'	Different scalabilities value (SNR, spatial, temporal)
18	$(2d_m + 1)^2$	Motion vector candidates
19	$[d_x, d_y]^T$	Minimize the existing block's distortion measure

Conflicts of Interest

The authors declare no conflict of interest

Author Contributions

Agus Purwadi; conception, approach, software, writing - first draft. Suwadi; data curation, visualization, investigation. Wirawan: resources, validation, supervision. Esa Prakasa: validation, supervision.

Acknowledgments

The author would like to thank the Data and Information Science Research Center, Electronics and Informatics Research Institute, National Research and Innovation Agency (BRIN), Indonesia for the support provided in the form of partial funding for this research and it is part of dissertation research at Electrical Engineering Department, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia.

References

[1] S. Metkar and S. Talbar, "Motion Estimation Techniques for Digital Video Coding", *SpringerBriefs in Applied Sciences and Technology*, 2013, doi: 10.1007/978-81-322-1097-9.

[2] I. Chakrabarti, K. N. S. Batta, and S. K. Chatterjee, "Motion Estimation for Video Coding", *Vol. 590. in Studies in Computational Intelligence*, Vol. 590, 2015, doi: 10.1007/978-3-319-14376-7.

[3] P. Helle *et al.*, "A Scalable Video Coding Extension of HEVC", In: *Proc. of 2013 Data Compression Conf*, Snowbird, UT, pp. 201-210, 2013, doi: 10.1109/DCC.2013.28.

[4] Z. Shi, X. Sun, and F. Wu, "Spatially Scalable Video Coding For HEVC", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 22, No. 12,

pp. 1813-1826, 2012, doi: 10.1109/TCSVT.2012.2223031.

[5] K. Yang, S. Wan, Y. Gong, Y. Yang, and Y. Feng, "Temporal distortions in H.265/HEVC", In: *Proc. of 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Jeju, South Korea, pp. 1-4, 2016, doi: 10.1109/APSIPA.2016.7820855.

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

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

[8] T. Hinz *et al.*, "An HEVC extension for spatial and quality scalable video coding", *presented at the IS&T/SPIE Electronic Imaging, A. Said, O. G. Guleryuz, and R. L. Stevenson, Eds.*, Burlingame, California, USA, p. 866605, 2013, doi: 10.1117/12.2009393.

[9] A. Barjatya, "Block Matching Algorithms for Motion Estimation", *DIP 6620 Spring 2004 Final Project Paper*, Utah State University, Logan, Utah, p. 1-5, 2004.

[10] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video processing and communications*, Prentice-Hall signal processing series, Upper Saddle River, N.J: Prentice Hall, 2002.

[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, 2012, doi: 10.1109/TCSVT.2012.2221191.

[12] ITU-T H.265 and ISO/IEC 23008-2, Series H: Audiovisual and Multimedia Systems Infrastructure of audiovisual services – Coding of moving, *High efficiency video coding*, ITU-T Recomendation H.265, 2019.

[13] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 11, No. 3, pp. 301-317, 2001, doi: 10.1109/76.911157.

[14] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block Partitioning Structure in the HEVC Standard", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 22, No. 12, pp. 1697-1706, 2012, doi: 10.1109/TCSVT.2012.2223011.

[15] V. Sze, M. Budagavi, and G. J. Sullivan, Eds., "High Efficiency Video Coding (HEVC): Algorithms and Architectures", *Integrated Circuits and Systems*, 2014, doi: 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, 2021, doi: 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", In: *Proc. of Green and Smart Technology 2016*, pp. 53-57, 2016, doi: 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, 2015, doi: 10.1016/j.image.2015.09.015.

[19] F. Zhai, R. Beryy, T. N. Pappas, and A. K. Katsaggelos, "A rate-distortion optimized error control scheme for scalable video streaming over the Internet", In: *Proc. of 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, Baltimore, MD, USA, p. II-125, 2003, doi: 10.1109/ICME.2003.1221569.

[20] D. Rüfenacht, "Novel Motion Anchoring Strategies for Wavelet-based Highly Scalable Video Compression", *in Springer Theses*, Singapore: Springer Singapore, 2018, doi: 10.1007/978-981-10-8225-2.

[21] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD curves", In: *Proc. of ITU-T SG 16/Q.6 13th VCEG Meeting*, Austin, Texas, USA, document VCEG-M33, 2001.

[22] ITU-T COM 16–C 404 – E, *On the calculation of PSNR and bit-rate differences for the SVT test data*, Study Group 16 – Contribution 404, Japan, 2008.

[23] [1]S. Metkar and S. Talbar, Motion Estimation Techniques for Digital Video Coding. in

SpringerBriefs in Applied Sciences and Technology. India: Springer India, 2013. doi: 10.1007/978-81-322-1097-9.

[2] I. Chakrabarti, K. N. S. Batta, and S. K. Chatterjee, Motion Estimation for Video Coding, vol. 590. in Studies in Computational Intelligence, vol. 590. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-14376-7.

[3] P. Helle et al., "A Scalable Video Coding Extension of HEVC," in 2013 Data Compression Conference, Snowbird, UT: IEEE, Mar. 2013, pp. 201–210. doi: 10.1109/DCC.2013.28.

[4] Z. Shi, X. Sun, and F. Wu, "Spatially Scalable Video Coding For HEVC," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1813–1826, Dec. 2012, doi: 10.1109/TCSVT.2012.2223031.

[5] K. Yang, S. Wan, Y. Gong, Y. Yang, and Y. Feng, "Temporal distortions in H.265/HEVC," in 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Jeju, South Korea: IEEE, Dec. 2016, pp. 1–4. doi: 10.1109/APSIPA.2016.7820855.

[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. doi: 10.1109/ISRITI54043.2021.9702868.

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

[8] T. Hinz et al., "An HEVC extension for spatial and quality scalable video coding," presented at the IS&T/SPIE Electronic Imaging, A. Said, O. G. Guleryuz, and R. L. Stevenson, Eds., Burlingame, California, USA, Feb. 2013, p. 866605. doi: 10.1117/12.2009393.

[9] A. Barjatya, "Block Matching Algorithms For Motion Estimation," DIP 6620 Spring 2004 Final Project Paper, Utah State University, Logan, Utah, p. 1-5, 2004.

[10] Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video processing and communications. in Prentice-Hall signal processing series. Upper Saddle River, N.J: Prentice Hall, 2002.

[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, doi: 10.1109/TCSVT.2012.2221191.

[12] ITU-T H.265 and ISO/IEC 23008-2, Series H: Audiovisual And Multimedia Systems Infrastructure of audiovisual services – Coding of moving, "High efficiency video coding", ITU-T Recomendation H.265, 2019

[13] Weiping Li, "Overview of fine granularity scalability in MPEG-4 video standard," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 3, pp. 301–317, Mar. 2001, doi: 10.1109/76.911157.

[14] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block Partitioning Structure in the HEVC Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1697–1706, Dec. 2012, doi: 10.1109/TCSVT.2012.2223011.

[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. doi: 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, doi: 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. doi: 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, doi: 10.1016/j.image.2015.09.015.

[19] Fan Zhai, R. Beryy, T. N. Pappas, and A. K. Katsaggelos, "A rate-distortion optimized error control scheme for scalable video streaming over the Internet," in 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698), Baltimore, MD, USA: IEEE, 2003, p. II–125. doi: 10.1109/ICME.2003.1221569.

[20] D. Rüfenacht, Novel Motion Anchoring Strategies for Wavelet-based Highly Scalable Video Compression. in Springer Theses. Singapore: Springer Singapore, 2018. doi: 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] ITU-T COM 16–C 404 – E, "On the calculation of PSNR and bit-rate differences for the SVT test data", Study Group 16 – Contribution 404, Japan, April 2008

[23] Xiph.org Foundation, "Derf's test media collection", *Our collection of test sequences and clips for evaluating compression technology*, 2017, https://media.xiph.org/video/derf/

[24] Fraunhofer Heinrich Hertz Institute, *HEVC test model (HM)*, https://hevc.hhi.fraunhofer.de/HM-doc/

[25] F. Zhang and D. R. Bull, "Rate-Distortion Optimization Using Adaptive Lagrange Multipliers", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 29, No. 10, pp. 3121-3131, 2019, doi: 10.1109/TCSVT.2018.2873837.

[26] Q. Liu, L. Hao, L. Liu, T. Peng, "Fast Motion Estimation Algorithm for High Efficient Video Coding", In: *Proc. of 2018 IEEE 4th International Conference on Computer and Communications*, Chengdu, China, p 6-10, 2018.

[27] K. Yang, S. Wan, Y. Gong, Y. Yang, and Y. Feng, "Temporal distortions in H.265/HEVC", In: *Proc. of 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Jeju, South Korea, pp. 1-4, 2016, doi: 10.1109/APSIPA.2016.7820855.

[28] R. Bailleul, J. De Cock, and R.Van De Walle, "Fast mode decision for SNR scalability in SHVC digest of technical papers", In: *Proc. of 2014 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, pp. 193-194, 2014, doi: 10.1109/ICCE.2014.6775968.