Journal of Engineering Science and Technology
ICIST2022, Vol. 18, No. 1 (2023) 23 - 35


# LOW COMPLEXITY VERSATILE VIDEO CODING (VVC) FOR LIVE VIDEO COMMUNICATION

ALEXANDER V. BUKIT[1,2,*], SUWADI[1], WIRAWAN[1]

[1]Dept. of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2]Dept. of Informatics Engineering, Indonesian Naval Technology College, Surabaya, Indonesia
*Corresponding Author: alex.v.bukit@sttal.ac.id

**Abstract**

VVC has a better coding efficiency relative to predecessor standards (HEVC and AVC), but also achieves increased complexity. VVC standardization committee seeks to achieve more effective compression, particularly for 4K and 8K contents by reducing the bitrate of compressed videos by 50% while maintaining the same level of objective and subjective quality. However, these tools negatively deliver the computational cost of the VVC. Because VVC requires a fairly large delay in video encoding, it is necessary to optimize the encoding process to obtain an optimal trade-off between efficiency and complexity of the coding process. To reduce encoding time, we propose an optimization method for VVC in this paper which can achieve VVC usage in live video communication, which requires as little coding time as possible to reduce delay time. Accordingly, we formulate an optimization for VVC usage conditions in live video communication at low bitrate and low resolution CIF (352×240) and QCIF (176×120). Experiment results show that this paper proposed method can reduce coding time (t %) average up to -67.51% through setting the initial VVC parameters with insignificant quality degradation.

Keywords: Low complexity, Low resolution, Versatile video coding, Video compression.

## 1. Introduction

The growth in the use of video applications in recent years has experienced a very large increase, in almost all aspects of daily life using video media. This of course will increase in bandwidth requirements for video media transmission and storage capacity. Versatile Video Coding (VVC) [1] is a recent video coding standard, ISO/IEC Moving Picture Experts Group (MPEG) and The International Telecommunication Union (ITU) and then the video standard finalized by the Joint Video Experts Team (JVET), a shared effort of ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG). Compared to its predecessor, H.265/HEVC, the new codec offers bit rate reductions of about 50% on average [2-9].

Media Video produces a fairly large file size because it handles moving images and sound. Along with the trend of increasing the video resolution used which is intended to improve the experience of enjoying a better video, resulting in a much larger video size. With comparable image quality, VVC can boost coding efficiency by around 50%. VVC, however, has a complexity that is 18 times greater than HEVC [10-16]. The partitioning mode, quad-tree with nested multi type tree (QTMT) concentrated more than 95% of the overall time [17]. Based on this, video encoding technology is needed to be able to reduce the size of the video without significantly reducing the quality so that transmission and storage can be done more economically and efficiently. Faced with these needs, the development of video encoding technology is getting better, so that it can provide higher compression efficiency and can save storage space and bandwidth of video transmission media. As rate control and coding time became one of the most important in video coding, there existed many researches towards it. Mourad et al. [18] designed a low complexity method on versatile video coding (VVC) for Low Bitrate Applications to achieve better bitrate in VVC standard. Abdoli et al. [19] designed Short Distance Intra Prediction for Screen Content to achieve better intra prediction in screen content of VVC standard.

While the VVC standards primarily focus on video content of HD or UHD including the formats of 3840 × 2160 (4K) and 7680 × 4320 (8K). Using Standard Definition (SD) video (480p or 360p) to reduce delay coding time and stabilize bandwidth-limited networks, VVC is also necessary for live video communication. Because VVC has a complexity and huge coding time, VVC is not efficient for live video communication. In our paper, we propose an optimization method for VVC which can achieve VVC usage in live video communication which requires as little coding time as possible to reduce delay time. Accordingly, we formulate an optimization for VVC usage conditions in live video communication at low bitrate and low resolution CIF (352×240) and QCIF (176×120). Although HD video has become the standard video on the Internet, low-resolution contents are still used, particularly in 480p and 360p [18-26]. Experiment results show that our proposed method can reduce coding time (t %) average up to -67.51% through setting the initial VVC parameters with insignificant quality degradation.

## 2. Related Works

The VVC optimization and modification has already been largely addressed in the literature. Numerous approaches, in particular, have focused on simplifying the Rate Distortion Optimization (RDO) process. Focusing on reducing the process of Intra mode candidates, Ma et al. [27] uses coding information of the neighbouring/adjacent blocks depth decision and Coding Unit (CU) size. An algorithm for immediate determination of the CU size was put out [28].

All previous approaches modify the non-normative encoding process algorithmically. In this paper, we specifically consider the usefulness of optimization of the new coding tools in VVC standard. The use for HD and UHD may not be optimum for live video communication, so some of VVC coding tools will be disabled, having minimal effects on compression efficiency. We formulate an optimization for VVC usage conditions in live video communication at low bitrate and low resolution CIF (352×240) and QCIF (17×120), so that it can reduce complexity of VVC standard.

The remainder of this paper is as follows. In Section 3, Versatile Video Coding (VVC). A brief overview of VVC standard's technical features and its coding tools. In Section 4, Proposed Methodology, and then Section 5, Results and Discussion. Finally, in Section 6 are the conclusions.

## 3. Versatile Video Coding (VVC)

VVC is a recent video coding standard developed by JVET, which aims to improve compression performance over existing HEVC standards significantly with the goal of saving 50% bitrate. Compared to the previous standard, the quality of video compression standard has increased. For instance, in the development of VVC, the standard will reduce the cost of storage media and video transmission while maintaining or enhancing video quality. This increase can reduce the cost of storage media and video transmission for video data as well as enhance the video quality of television broadcasts that regularly employ the same channel size. The increased compression efficiency of VVC increases complexity and cost. There are many different types of complexity, such as computational complexity, memory bandwidth and memory requirements (local and global) [29-35]. The method used in the VVC, similar to AVC and HEVC and other state of the art video codecs, is a hybrid video coding method. Hybrid video coding focuses on three forms: block splitting, inter and intra prediction, and conversion process. In these processes, for encoding, the input image is first divided into fixed size (up to 128 x 128 luminance samples) square coding tree units (CTUs). High-level extensions allow to further subdivide the image based on the basic subdivision and define logical sub-image areas such as tiles, slices, and sub-images. Using a flexible partitioning scheme, each CTU will be subdivided into some rectangular coding units (CUs) [36]. Performing video compression calculations requires far more decision-making processes, resulting in an increase of the processor load to finish the video encoding process. In VVC it is possible for QuadTree with a nested multi-type tree (MTT) that is used for each CTU and a binary and ternary split structure to be used (QTBTTT). The quadtree leaf node can then be partitioned vertically or horizontally by a binary or turn around partitioning structure. Then the square or rectangular coding units (CU) are used for residual coding and prediction.

From Fig.1, VVC will processes the candidate of motion values for block Y(u,v) as the current view block, with (u,v), that will be assumed as the location of block's center, and d is an associated disparity vector used to obtain the sample point of corresponding reference (uR, vR) = (u + d, v). From this block sample location, the reference block YR(uR, vR) pointed to the location of the prediction unit (uR, vR) in the reference view.

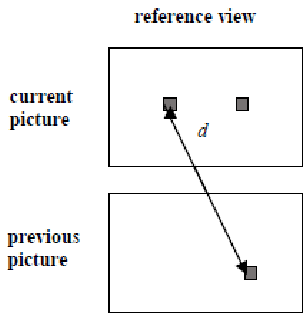


**Fig. 1. Motion vector correspondences using the disparity vector d between a block in the current picture and the previously encoded reference view.**

Some versions of the VVC software exist open-source: VVC test model (VTM) 2.0 until VTM 14.2. The VTM source code is available on the GitHub website [37]. For this test, we used VVC Test Model (VTM) 14.2 that has a 65 angle intra prediction mode including planar and DC mode. VTM 14.2 expands the list of most likely modes (MPM) to six candidates. Set up three new ways to intra prediction of VTM 14.2 blocks: (1) Intra Sub Partitions (ISP), which is vertically or horizontally divided by Luma CB in two or four subpartitions, (2) Cross component linear model (CCLM), the chrome sample has a linear model based on the reconstructed luminance sample of the same CU, (3) Matrix-based intra prediction (MIP) takes a series of reconfigured neighbouring samples. File organization structure of VTM project:

- Build file: the project file NextSoftware.sln containing VTM;
- cfg file: It contains the VTM encoding configuration file .cfg, which can be opened and edited directly with a text tool;
- Source file: It is the C++ source file of VTM, including the source file of encoder and decoder;
- bin file: Contains the compilation results of the VTM project, EncoderApp.exe and DecoderApp.exe, etc.
- DecoderAnalyserApp:
- DecoderApp: Decoder application function
- EncoderApp: Encoder application function
- CommonLib: Common library functions for encoders and decoders
- DecoderLib: Decoder library function
- EncoderLib: Encoder library function

After intra-picture prediction process, all of sequence inter-picture prediction remaining pictures may be used, using the coding modes of temporally-predictive to get the prediction of a new prediction block (PB), employing the image data from prior reference pictures, and using motion compensated prediction. To improve accuracy, motion prediction will be performed at sub-CU level. VTM supports SbTMVP (Sub-Pu Temporal Motion Vector Prediction) and AFFine motion compensation prediction (AFF) with irregular motions such as zoom in/out and rotation.

Inter-picture prediction process is done by selecting reference picture and motion vector (MV), motion data comprising to be applied in the prediction of the samples of each block by applying motion compensation (MC) using the mode decision data and MV, which will be transmitted as additional information. The motion vector of the current block usually correlates with the motion vector of the current image or adjacent blocks in the previously encoded image. This is due to the likelihood that adjacent blocks will correspond to identical moving objects that move similarly, and the object's movement is unlikely to suddenly alter over time. As a result, using the motion vectors of adjacent blocks as predictors reduces the magnitude of the difference in the motion vectors of the signal. MVPs (motion vector predictor) are usually derived from motion vectors that have already been decoded, either from spatially adjacent blocks or from temporally adjacent blocks in a co-located image [38].

The new concepts of VVC are the bi-prediction mode which extends beyond of the simple weighted averaging method, by utilizing Generalized Bi-prediction (GBI) as five predefined weights, and until pixel level motion refinement BDOF (Bidirectional Optical Flow). Triangular prediction (Triang), Combined Inter and Intra Prediction (CIIP) will merge with Motion Vector Difference (MVD) scheme (MMVD and Symmetric MVD (SMVD).

The quad-tree with nested multi-type tree (QTMT) structure eliminates the differences among the coding unit (CU), transform unit (TU) and prediction unit (PU), and only accepts the CU concept. The CU consists of a rectangular and square in QTMT structure. The coding tree unit (CTU) is categorized into quad-tree (QT), the size is 16×16 until 128×128, and the CTU size is 128×128. The minimum quad-tree size is 16×16 and 128×128 is the maximum quadtree size.

The Dependent Quantization (DepQuant), where the value of reconstruction for a transform coefficient will be adapted on the transform coefficient value in the reconstruction order.

**a.** Intra-Picture Prediction: Adopted from HEVC, advanced intra-picture prediction technique in VVC has more angles than HEVC, including the DC and planar modes [39].

- MIP (Matrix Based Intra Picture Prediction): can be considered as a simpler neural network based variant in-image prediction. The MIP mode matrix's selected index is signalled in truncated type binary code. Commonly, MIP only applies to luminance samples in most common use case, but this method can also be used to chroma samples for chroma sampling 4: 4: 4, with CST disabled [40].
- Intra Sub Partition (ISP) Mode: in this mode will split the CU luma block horizontally or vertically into 2 until 4 sub-partitions, facilitating independent process of prediction and transformation with each sub-partition sharing the information of coding mode value.
- Cross-Component Linear Model (CCLM): uses corresponding reconstructed luma samples to predict chroma component samples. The Cross-Component Linear Model parameters will be derived by minimizing the error of regression between adjacent chroma and luma samples [41].

**b.** Inter-Picture Prediction: single-MV un-prediction, referring an image from the data of reference pictures that previously decoded, and bi-prediction is used for interpicture prediction in VVC [42] For more effective process in prediction, representation, and motion-compensation control information coding, VVC includes various new coding methods. The improvement of coding motion information: In VVC, utilize the information of motion from spatially and temporally adjacent blocks, the advanced and the merge mode of MV prediction (AMVP) for motion parameters coding and prediction in extended using improved predictors and inter-picture prediction. For the merge mode, it will enable MV differences (MVDs), and this mode will give more flexibility to MVD signalling [43]. Advances in the compensation of CU-level motion: By incorporating more flexible the prediction signals weighting, VVC improves CU-level motion compensation. VVC can be processed as combination of merge mode with signalling of biprediction weights and intra-picture prediction at the CU-level. Refined motion compensation based on subblock: VVC improve the technologies that depict motion in better granularity, improve fractional sample motion compensation and MV precision [44].

- Symmetric MVD (SMVD): this mode is used to code the information of motion in bi-prediction mode, that when it is feasible will use a shortcut process. AMVP and MVD parameters is 0 and 1, signalled in SMVD mode.
- Adaptive MV Resolution (AMVR): The MV resolution is used at the CU level. This mode can provide a better trade-off between prediction quality and MV overhead bits. Switchable interpolation filter (SIF) is another name for this aspect of AMVR.
- Merge with MVD (MMVD): In this mode, the MVD is the indicator of the distance and selected direction.
- Combined Intra/Inter-Picture Prediction (CIIP): Choosing an intra-picture prediction mode or not, based on whether the left and above adjacent CUs are coded. At the final prediction, CIIP mode will be as weighted combination of a prediction value on planar mode intra-picture and a merge mode inter-picture.
- Subblock Based Temporal MV Prediction (SBTMVP): Two major steps are used in SBTMVP derivation: a) create a motion vector (DV) for the current CU, and b) find the information of motion for each subblock from the motion that DV identifies.
- Affine Motion: Either a 6-parameter or 4-parameter model can be used for CU-level affine motion. The top-left and top-right corners of the CU serve as the MVs of two control points for the 4-parameter model, while the top-left, bottom-left, and top-right corners serve as the MVs of three control points for the 6-parameter model.
- Decoder MV Refinement (DMVR): With a mirrored MV offset, DMVR searches candidate MVs in list 0 and list 1 that are close to the original MVs. The sample of fractional MV offset refinement and MV offset search make up the search process.

**c.** Transforms and Quantization: By combining extended transforms with refined quantization and residual coding, VVC improves energy compaction of the prediction residual.

- Low Frequency Non-Separable Transform (LFNST): This mode is selected on the intra mode, then an index will be explicitly signalled at each CU to select the block of LFNST. If blocks have height or width 4, LFNST inverse transform process has 16 outputs and 48 outputs for larger blocks.
- Subblock Transform (SBT): select the sub-partition when coding an inter-predicted CU residual block and leave the rest un-coded.

**d.** In-Loop Filters: Before using the pictures as references and for output on later motion compensated prediction, the reconstructed video signal is first subjected to improve and new signal-enhancing in-loop filters in VVC. The inverse luma mapping component of a new luma mapping with chroma scaling tool is applied before all other in-loop filters in this [45].

- Luma Mapping with Chroma Scaling (LMCS): This mode is to select the dynamic range to match the encoder's forward mapping function, using the input signal of a luma inverse mapping function and it will be applied to the loop filtering and then reconstructed video before.
- Adaptive Loop Filter (ALF): this mode will apply a process of spatial filtering to improve the result of video signal. This enables an encoder by signalling and selecting appropriate clipping parameters, to take the value of similarity between adjacent and current samples.

At the last stage of video encoding (and the first stage of video decoding), VVC performs entropy coding which is a type of lossless compression scheme. This scheme will employ statistical values to compress data so the bit's number will be used to represent the data proportional to the data probability. For example, infrequently used characters are each represented by many bits while frequently used characters are each represented by a few bits. The VVC method for entropy coding is called CABAC (Context Adaptive Binary Arithmetic Coding). H.265 and H.264 both made use of these formats. For great coding efficiency, context modelling estimates the probability of the bins. The three main functions performed by CABAC are binarization, context modelling, and arithmetic coding [46].

## 4. Proposed Methodology

In this study, the aim is to obtain the BDrate% value and decrease the amount of time compared to the initial value (when all variables are active) to complete coding with parameter variables that are turned off alternately to get a comparison of the increase in BDrate% and decrease in the amount of time (t%). The parameter variables in VVC tested in this work are shown in Table 1.

**Table 1. Parameter variables in VVC tested in this work.**

| Parameter Variable | Description |
|---|---|
| **CCLM** | Chroma-prediction based on linear-model |
| **MRL** | Multiple-Reference Line intra-prediction |
| **MIP** | Matrix based Intra-prediction |
| **ISP** | Intra Sub Partitions |
| **CIIP** | Combined Inter and Intra prediction |
| **SbTMVP** | Sub Pu-Temporal Motion-Vector-Prediction |
| **AFF** | AFFine inter-motion compensation |
| **MMVD** | Merge-with-MVD |
| **SMVD** | Symmetric-MVD |
| **DMVR** | Decoder Side Motion Vector Refinement |
| **LFNST** | Low-Frequency Non-Separable Transform |
| **LMCS** | Luma Mapping with Chroma Scaling |
| **ALF** | Adaptive Loop Filter |

VTM 14.2 is used in our experiments, on a PC with Intel ® Core ™ i7 – 67000T cores @ 2.80 GHz running under Linux Ubuntu 20.04.4 LTS. Parameter variable for setting to be measured on VTM, in video encoding is created into encoder_randomaccess_vtm.cfg file. It contains the VTM encoding configuration

file which can be opened and edited directly with a text tool. The parameters of the cfg file need to be configured, including the parameter configuration of the VTM project itself and the parameter configuration of the test sequence, in the ./cfg directory. The file contains the configuration parameters at the time of running the video encoding. The parameters will be a reference for the implementation of the video coding process using the VTM application.

In this video encoding, low resolution video and configuration parameters are used which are inputted to VTM to get the value of delay time (time required to complete encoding) by changing the disabled configuration parameters, faced with the complexity possessed by VVC, excellent compression capabilities and requirements. small delay time to meet the needs of real time video communication.

Furthermore, the sequences used in this study are those with low resolution characteristics of CIF (352×240) and QCIF (176×120). The sequences used in this study are as listed in Table 2. Real time video communication is widely used nowadays such as video conferencing, social media, chat and television over the internet. In this system, users can view, record and edit video content using time-critical infrastructures. The video resolution that is mostly used in real time video communication is low resolution in order to minimize delay time caused by the need for coding time on the VVC standard.

**Table 2. Sequences used in this study.**

| No. | Sequence |
|---|---|
| **1.** | news_cif.y4m |
| **2.** | news_qcif.y4m |
| **3.** | bus_cif.y4m |
| **4.** | bus_qcif.y4m |
| **5.** | foreman_cif.y4m |
| **6.** | foreman_qcif.y4m |

## 5. Result and Discussion

In this study, sequences with low resolution 352×240 were used, news_cif.y4m, bus_cif.y4m, foreman_cif.y4m and resolution 176×120 were used, news_qcif.y4m, bus_qcif.y4m, foreman_qcif.y4m [47]. We use the Bjontegaard Delta Rate ($BD_{Rate}$) [48] to evaluate the test result and t% is the decreasing coding time in percent compared with the anchor test result.

From Table 3 and 4, it can be seen that the results of the tests that have been carried out obtained from the results of $BD_{Rated}$ and decreased coding time (t %) when disabling one of the VTM input parameter variables. AFF, MMVD, BDOF and ALF have a significant value in decreased coding time (t%) as shown in highlighted cells. Next, we will test for the combination of these tools and the result as shown in Table 5. The comparison of t(anchor) versus t(method) and BDrate(anchor) versus BDrate(method) is shown in Fig. 2.

At the combination tools test result, the average reduction in BDRate% was 2.19 and the decrease in t% was -67.51, so that it can benefit from a fairly large decrease in t% compared to the decrease in BDRate which is quite small, so it can be applied to real time online video communication, which requires as little coding time as possible to reduce delay time. Based on our knowledge, this is the first work on literature that proposed using VVC at low complexity for live video communication. The test result t% is the reduction of coding time in percent compared with the anchor test result. These results are obtained by disabling tools for the initial VVC parameters. The combination of tools will provide the optimal complexity video coding reduction. The complexity amount reduction will be varying due to the sequence's temporal and spatial properties. It has the same thing happened in all tests, and it is hoped that these results can apply in general to all video coding using the VTM standard.

**Table 3. Test results on sequences.**

| Tools | news_cif | | bus_cif | | foreman_cif | |
|---|---|---|---|---|---|---|
| | $BD_{Rate}$% | t % | $BD_{Rate}$% | t % | $BD_{Rate}$% | t % |
| **CCLM** | 2.459 | 1.87 | 1.25 | 6.58 | 0.304 | -0.713 |
| **MRL** | 0.82 | 1.62 | 3.76 | 0.19 | 0.609 | 0.174 |
| **MIP** | 6.39 | 1.16 | -4.039 | -0.16 | 0,61 | 0.54 |
| **ISP** | 2.30 | 1.71 | -6.69 | -3.38 | 0.01 | -0.53 |
| **CIIP** | 4.92 | -0.68 | 1.39 | -4.56 | -0.30 | -0.76 |
| **SbTMVP** | 0.08 | -2.13 | -0.7 | -1.6 | -1.37 | 2.17 |
| **AFF** | 2.13 | -22.22 | 0.14 | -21.86 | 0.46 | -23.31 |
| **MMVD** | 0.33 | -6.79 | -3.34 | -9.46 | -6.09 | -4.46 |
| **SMVD** | 0.01 | -0.6 | -0.56 | -2.07 | 0.01 | 0.344 |
| **BDOF** | -0.16 | -3.42 | -0.56 | -4.81 | -1.37 | 0.51 |
| **DMVR** | -1.8 | -2.68 | -1.95 | -2.37 | -8.68 | 14.82 |
| **AMVR** | 0.01 | -3.42 | -0.14 | -3.46 | 0 | 13.06 |
| **LFNST** | 0.01 | 1.53 | -4.74 | -4.13 | 0.01 | 16.13 |
| **LMCS** | 0.01 | 0.63 | 0.01 | 1.14 | 0.01 | -0.73 |
| **ALF** | 0.01 | -39.96 | 0.01 | -41.44 | 0.01 | -30.37 |

**Table 4. Test results on sequences with qcif format.**

| Tools | news_qcif | | bus_qcif | | foreman_qcif. | |
|---|---|---|---|---|---|---|
| | $BD_{Rate}$% | t % | $BD_{Rate}$% | t % | $BD_{Rate}$% | t % |
| **CCLM** | 3.16 | 4.67 | -0.12 | 2.46 | -0.46 | -1.43 |
| **MRL** | 0.01 | -2.00 | -2.03 | -5.45 | 0.01 | 10.38 |
| **MIP** | -2.73 | -3.88 | -4.06 | 7.90 | -4.92 | -0.69 |
| **ISP** | 0.86 | 6.33 | 0.20 | 10.49 | -1.23 | 6.35 |
| **CIIP** | 2.73 | -2.97 | 1.65 | -1.43 | 0.77 | -2.94 |
| **SbTMVP** | -0.43 | -1.35 | 0.63 | 0.39 | 0.01 | -0.03 |
| **AFF** | -1.58 | -23.40 | 0.89 | -22.85 | 0.62 | -20.35 |
| **MMVD** | 0.01 | -11.08 | 2.92 | -6.49 | 1.54 | -9.08 |
| **SMVD** | 0.01 | -3.98 | 3.17 | 1.86 | 1.69 | 2.90 |
| **BDOF** | 0.72 | -8.66 | 3.03 | -4.87 | 0 | -4.67 |
| **DMVR** | -0.29 | -5.50 | -4.95 | -3.37 | 0 | -3.50 |
| **AMVR** | -1.43 | -6.09 | -0.89 | -1.45 | 0.01 | -3.83 |
| **LFNST** | -3.44 | 0.04 | -9.01 | -6.26 | 4.91 | 3.88 |
| **LMCS** | 0.01 | -2.73 | 0.01 | 2.18 | 0.01 | -0.20 |
| **ALF** | 0.01 | -37.16 | 0.01 | -34.98 | 0.01 | -36.25 |

**Table 5. Test results on combination variables (AFF, MMVD, BDOF and ALF).**

| news_cif | | news_qcif | |
|---|---|---|---|
| $BD_{Rate}$% | t % | $BD_{Rate}$% | t % |
| 1.43 | -71.61 | 2.95 | -63.40 |
| **$BD_{Rate}$ % Average** = 2.19 | | | |
| **t% Average** = -67.51 | | | |

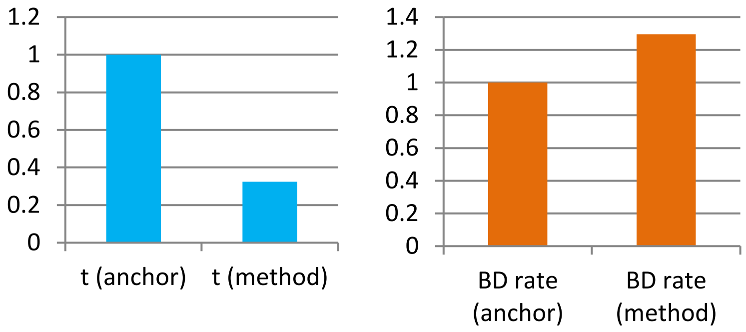


**Fig. 2. Test results on combination variables (AFF, MMVD, BDOF and ALF) BDrate and t of anchor vs proposed method.**

## 6. Conclusions

The experimental validation of VVC optimization for live communication video has been reported. This research is aimed at optimizing low resolution video coding in cif and qcif formats. The validation using VTM 14.2 (VVC test mode) standard to code low resolution cif and qcif video for the use in real time video communication. The optimization can be achieved by disabling one of VTM coding input parameter variables, so that it can reduce the coding time (t%) for -67.51 which is quite significant compared to the decreasing BDRate level which is quite small that it can be ignored.

The test results show that the results are quite good by reducing the time needed to complete the coding required for video communication in real time. It is hoped that the same results can also be obtained when used for other sequences so that they can be applied in general.

## Acknowledgment

Authors would like to acknowledge support provided by the Electrical Engineering Department, Faculty of Intelligent Electrical and Informatics Technology, ITS on research facilities and financial support during completion of this research.

## References

1. ITU-T and ISO/IEC JTC 1, (2020). Versatile video coding. *Rec. ITU-T H.266 and ISO/IEC 23090-3 (VVC).*
2. Wien, M.; and Baroncini, V. (2020). Report on VVC compression performance verification testing in the SDR UHD random access category. *document JVET-T0097, Joint Video Experts Team (JVET).*
3. CanMert, A.; Kalali, E.; and Hamzaoglu, I. (2018). A low power versatile video coding (VVC) fractional interpolation hardware. *Proceedings of the 2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, Porto, Portugal, 43-47.
4. Yoon, Y.-U.; and Kim, J.-G. (2022). Activity-based block partitioning decision method for versatile video coding. *Electronics*, 11(7), 1061.
5. Azgin, H.; Kalali, E.; and Hamzaoglu, I. (2019). An efficient FPGA implementation of versatile video coding intra prediction. *Proceedings of the*

*2019 22nd Euromicro Conference on Digital System Design (DSD)*, Kallithea, Greece, 194-199.

6. Silva, G.G.; Siqueira, ´I.G.; Grellert, M. ; and Diniz, C.M. (2021). Approximate hardware architecture for the interpolation filter of versatile video coding. *Journal of Integrated Circuits and Systems*, 16(2), 1-8.
7. Luo, D.; Xiong, S.; Ren, C.; Sheriff, R.E.; and He, X. (2022). Fusion-based versatile video coding intra prediction algorithm with template matching and linear prediction. *Sensors*, 22(16), 5977.
8. He, G.; Xu, K.; Wu, C.; Ma, Z.; Wen, X.; and Sun, M. (2022). Hybrid video coding scheme based on VVC and spatio-temporal attention convolution neural network. *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 1791-1794.
9. Farhat, I.; Hamidouche, W.; Grill, A.; Ménard, D.; and Déforges, O. (2021). Lightweight hardware transform design for the versatile video coding 4K ASIC decoders. *IEEE Transactions on Consumer Electronics*, 67(4), 329-340.
10. VTM software. Retrieved October 5, 2022, from https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware VTM/
11. Nguyen, T.; Bross, B.; Schwarz, H.; Marpe, D.; and Wiegand, T. (2020). Residual coding for transform skip mode in versatile video coding. *Proceedings of the 2020 Data Compression Conference (DCC)*, Snowbird, UT, USA, 83-92.
12. Kim, S.; Chen, J.; Ye, Y. (2018). Algorithm description for versatile video coding and test model 2 (VTM). *Document JVET-K1002, Ljubljana, Slovenia*.
13. Fan, K.; Wang, R.; Wang, Y.; Li, G.; and Gao, W. (2017). Improved intra boundary filters for HEVC. *Proceedings of the 2017 Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL, USA 1-4.
14. Park, D.; Kim, J.-G.; Lee, J.; and Kang, J. (2019). Simplification of BCW in versatile video coding (VVC). *Proceedings of the Korea Society of Broadcast Engineers Conference*, 22-23.
15. Khairat, A.; Nguyen, T.; Siekmann, M.; Marpe, D.; and Wiegand, T. (2014). Adaptive cross component prediction for 4: 4: 4 high efficiency video coding. *Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*, Paris, Franc, 3734-3738.
16. Cao, X.; Peng, X.; Lai, C.; Wang, Y.; Lin, Y.; Xu, J.; Liu, L.; Zheng, J.; He, Y.; Yu, H.; and Wu, F. (2011). CE6.b1 report on short distance intra prediction method. *Document JCTVC-E278, Geneva, Switzerland.*
17. Tissier, A.; Mercat, A.; Amestoy, T.; Hamidouche, W.; Vanne, J.; and Menard, D. (2019). Complexity reduction opportunities in the future VVC intra encoder. *Proceedings of the 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP).* Kuala Lumpur, Malaysia, 1-6.
18. Aklouf, M.; Leny, M.; Dufaux, F.; and Kieffer, M. (2019). Low complexity Versatile Video Coding (VVC) for low bitrate applications. *Proceedings of the 2019 8th European Workshop on Visual Information Processing (EUVIP)*, Rome, Italy, 22-27.
19. Abdoli, M.; Henry, F.; Brault, P.; Duhamel P.; and Dufaux, F. (2018). Short-distance intra prediction of screen content in versatile video coding (VVC). *IEEE Signal Processing Letters*, 25(11), 1690-1694.

20. Abdoli, M.; Henry, F.; Brault, P.; Dufaux, F.; and Duhamel, P. (2019). Transform coefficient coding for screen content in versatile video coding (VVC). *Proceedings of the ICASSP - IEEE 19th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 1792-1796.
21. Abdoli, M.; Henry, F.; Brault, P.; Dufaux, F.; Duhamel, P. and Philippe, P. (2019). Intra block-DPCM with layer separation of screen content in VVC. *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP),* Taipei, Taiwan, 3162-3166.
22. Topiwala, P.; Krishnan, M.; and Dai, W. (2018). Performance comparison of VVC, AV1 and HEVC on 8-bit and 10-bit content. *Applications of Digital Image Processing XLI,* 10752, 305-314.
23. De-Luxán-Hernández, S.; George, V.; Ma, J.; Nguyen, T.; Schwarz, H., Marpe, D.; and Wiegard, T. (2019). An intra sub partition coding mode for VVC. *Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP),* Taipei, Taiwan, 1203-1207.
24. Zhao, X.; Kim, S.-H.; Zhao, Y.; Egilmez, H.E.; Koo, M.; Liu, S.; Lainema, J.; and Karczewics, M. (2021). Transform coding in the VVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10), 3878-3890.
25. Schwarz, H.; Coban, M.; Karczewicz, M.; Chuang, T.-D.; Bossen, F.; Alshin, A.; and Wiegand, T. (2021). Quantization and entropy coding in the versatile video coding (VVC) standard. *IEEE Transactions on Circuits and systems for Video Technology*, 31(10), 3891-3906.
26. Versatile Video Coding. (2020). D*ocument ITU-T Rec. H.266 and ISO/IEC 23090-3, ITU-T and ISO/IEC.*
27. Zhao, L.; Zhang, L.; Ma, S.; and Zhao, D. (2011). Fast mode decision algorithm for intra prediction in HEVC. *Proceedings of the 2011 Visual Communications and Image Processing (VCIP), Tainan*, Taiwan, 1-4.
28. Shen, L.; Liu, Z.; Zhang, X.; Zhao, W.; and Zhang, Z. (2012). An effective CU size decision method for HEVC encoders. *IEEE Transactions on Multimedia,* 15(2), 465-470.
29. Huang, Y.-W.; An, J.; Huang, H.; Li, X.; Hsiang, S.-T.; Zhang, K.; and Chubach, O. (2021). Block partitioning structure in the VVC standard. *IEEE Transactions on Circuits Systems for Video Technology,* 31(10), 3818-3833.
30. Zhao, L.; Zhao, X.; Liu, S.; Li, X.; Lainema, J.; Rath, G.; and Racape, F. (2019). Wide angular intra prediction for versatile video coding. *Proceedings of the 2019 Data Compression Conference* (*DCC*), Snowbird, UT, USA, 53-62.
31. Zhao, L.; Zhao, X.; Li, S.; and Liu, X. (2018). CE3-related: Unification of angular intra prediction for square and non-square blocks. *JVET-L0279,* 3-12.
32. Filippov, A.; Rufitskiy, V.; Chen, J.; and Alshina, E. (2020). Intra prediction in the emerging VVC video coding standard. *Proceedings of the 2020 Data Compression Conference* (*DCC*), Snowbird, UT, USA, 367-367.
33. Zhao, X.; Seregin, V.; Said, A.; Zhang, K.; Egilmez, H.E.; and Karczewicz, M. (2018). Low-complexity intra prediction refinements for video coding. *Proceedings of the 2018 Picture Coding Symposium* (*PCS*), San Francisco, CA, USA,139-143.
34. Van der Auwera, G.; Heo, J.; and Filippov, A. (2018). CE3: Summary report on intra prediction and mode coding. *Document JVET-K0023, in the Proceeding of 11th Joint Video Exploration Team,* Geneva, Switzerland.

35. Bossen, F.; Sühring, K.; Wieckowski A.; and Liu, S. (2021). VVC complexity and software implementation analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10), 3765-3778.
36. Wieckowski, A.; Bross, B.; and Marpe, D. (2021). Fast partitioning strategies for VVC and their implementation in an open optimized encoder. *Proceedings of the 2021 Picture Coding Symposium (PCS)*, Bristol, United Kingdom, 1-5.
37. VTM 2.0 – VTM 14.2 Retrieved October 5, 2022, from https://github.com/fraunhoferhhi/vvenc.
38. Sze, V.; Budagavi, M.; and Sullivan, G.J. (2014). *High efficiency video coding (HEVC). Algorithms and architectures*. Integrated Circuit and Systems, Springer Cham Heidelberg New York Dordrecht London.
39. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; and Ohm, J.R. (2021). Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions On Circuits and Systems for Video Technology*, 31(10), 3736-3764.
40. Pfaff, J.; Filippov, A.; Liu, S.; Zhao, X.; Chen, J.; De-Luxán-Hernández, S.; and Van der Auwera, G. (2021). Intra prediction and mode coding in VVC. *IEEE Transactions On Circuits and Systems for Video Technology*, 31(10), 3834-3847.
41. Bossen, F.; Sühring, K.; Wieckowski, A.; and Liu, S. (2021). VVC complexity and software implementation analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10), 3765-3778.
42. Fu, T.; Zhang, H.; Mu, F.; and Chen, H. (2019). Fast CU partitioning algorithm for H.266/VVC intra-frame coding. *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, Shanghai, China,55-60.
43. Chen, Z.; and Pan, X. (2019). An optimized rate control for low-delay H.265/HEVC. *IEEE Transactions On Image Processing,* 28(9), 4541-4552.
44. Choi, Y.-J.; and Kim, B.-G. (2019). A review on motion estimation and compensation for versatile video coding technology (VVC). *Journal of Korea Multimedia Society*, 22(7), 770-779.
45. Pakdaman, F.; Adelimanesh, M.A.; Gabbouj, M.; and Hashemi, M.R. (2020). Complexity analysis of next-generation VVC encoding and decoding. *Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, 3134-3138.
46. Martínez-Rach, M.O.; Migallón, H.; López-Granado, O.; Galiano, V. and Malumbres, M.P. (2021). Performance overview of the latest video coding proposals: HEVC, JEM and VVC. *Journal of Imaging*,7(2): 39.
47. xiph.org (2022). Xiph.org Video Test Media [derf's collection]. Retrieved October 5, 2022, from https://media.xiph.org/video/derf/
48. tbr/bjontegaard_etro (2022). Retrieved October 5, 2022, from https://github.com/tbr/bjontegaard_etro