Evaluation of Tile-based Parallelism in VVC Encoders for 360-degree Videos

Éric Beltrão[1,2], Iago Storch[1], Luciano Agostini[1], Bruno Zatt[1], Daniel Palomino[1]

[1]Video Technology Research Group (ViTech), Federal University of Pelotas (UFPel)

[2]Federal Institute of Education, Science and Technology Farroupilha (IFFar)

{eric.beltrao, icstorch, agostini, zatt, dpalomino}@inf.ufpel.edu.br

Abstract. *This work evaluates the parallelism potential and the coding efficiency of Tile-enabled encoding of 360-degree videos for the Versatile Video Coding (VVC) standard. Six 4K conventional videos and eight 360-degree videos were tested with various tile configurations. The results showed that increasing the number of tiles led to speedup gains, with a maximum speedup of 15.52. However, coding efficiency, measured by BD-Rate, generally increases as the number of tiles increases.The results suggest that while both video types benefit from tile configurations, optimal configurations provide better speed gains and less degradation in coding efficiency, particularly in 360-degree videos.*

1. Introduction

Due to technological advances in the field of digital media, digital videos have become increasingly present in people's daily lives. Whether on mobile devices, televisions, or through computers, a large portion of the population shares digital videos and uses them to participate in virtual conferences or, at the very least, to watch video streaming. According to Sandvine's Global Internet Phenomena Report, digital video content accounted for 65.93% of all internet traffic in 2022, a significant 24% increase compared to 2021 [Sandvine 2023]. One of the main drivers of this growth is the increased use of video-on-demand services, such as Netflix and YouTube [Margetis et al. 2023]. Thus, efficient video compression is necessary to reduce the volume of data transmitted, thereby preventing overload on the transmission network or the storage media for such videos.

Besides conventional 2D videos, there is a growing popularity of 360-degree videos due to their ability to provide immersive and interactive experiences, allowing viewers to explore all angles of a scene. These videos are designed to be encoded using conventional coding standards, which already face significant efficiency and computational complexity challenges. However, the nature of 360-degree videos, which require higher resolutions to capture details in all directions, further intensifies these challenges. As a result, processing times and bandwidth demand increase, making the encoding of these videos even more complex than conventional videos.

The Versatile Video Coding (VVC) was released in July 2020, along with its reference software called VTM (VVC Test Model) in its version 10.0, which is considered the state-of-the-art standard among current video coding standards [Bross et al. 2021]. The VVC surpasses the High Efficiency Video Coding (HEVC) standard, achieving a 50% reduction in the bit rate of encoded videos while maintaining the same objective quality when compared to HEVC [Sullivan et al. 2012]. To achieve such a level of coding efficiency, several enhancements were made to the tools inherited from HEVC, and new tools were added to the encoding process. Due to these improved and new tools, an encoding following the VVC standard demands around eight times the encoding required by HEVC [Bross et al. 2021]. Hence, it is necessary to develop solutions to speed up the video encoding process to enable applications to use the VVC standard.

Modern encoders use complex strategies to improve coding efficiency, leading to large processing times. Nonetheless, most modern computing systems are composed of multiple processing cores. Under these circumstances, recent video coding standards have already introduced encoding tools designed to exploit parallelism in multicore CPUs, namely, Tiles, Slices, and Wavefront Parallel Processing (WPP) [Belememis et al. 2020]. Among these tools, tiles offer great scalability concerning the number of processing cores available [Wang et al. 2021]. In summary, tiles divide the video frame into rectangular regions (i.e., tiles) and break the data dependencies between these regions so that each tile can be independently encoded [Panagou et al. 2022]. Hence, exploring the parallelism opportunities enabled by Tiles on the VVC standard poses a great candidate for mitigating the long processing times required to encode 360-degree videos.

Several studies explore the impact of parallelization on video encoding. In HEVC, [Storch et al. 2020] evaluated tile division patterns to optimize encoding speedup, improving it by 40% with low compression efficiency loss. For VVC, [Amestoy et al. 2020] proposed a dynamic frame partitioning approach to minimize encoding time with minimal quality loss, while [Lee et al. 2023] analyzed the improvements of VVC over HEVC, highlighting its increased computational complexity. Specific methods for 360-degree videos have also been studied, such as FastInter360 [Storch et al. 2021], which reduces encoding time by 22.84% by optimizing interframe prediction, and [Jeong et al. 2020], which focuses on extracting optimized tiles for streaming, achieving a 24.81% bitrate reduction. However, among the solutions proposed in the literature, no approaches focus on tile parallelism in the encoding of 360-degree videos. Therefore, it is important to study tile-based parallelization and optimization techniques for improving video encoding efficiency, especially in high-resolution and immersive video scenarios.

The main goal of this work is to evaluate the parallelism potential for 360-degree videos, addressing aspects such as encoding efficiency and speedup based on tile parallelism. This is done by comparing a uniform tile segmentation approach with an optimal segmentation approach previously proposed for conventional videos.

2. VVC Standard and Parallelism Exploration

A large share of the coding efficiency gains of VVC comes from its highly flexible partitioning structures. A video frame encoded with the VVC standard is initially divided into a grid of square-shaped blocks called Coding Tree Units (CTUs), usually composed of 128×128 samples each [Wang et al. 2021]. These CTUs are further partitioned into Coding Units (CUs) following a hierarchical Multi-Type Tree (MTT) partitioning scheme, which extends the Quad-Tree (QT) structure from HEVC by incorporating Binary Trees (BTs) and Ternary Trees (TTs). This flexible partitioning allows for better adaptation to varying texture within a frame. Multiple encoding modes are tested for each partitioning, seeking the best coding efficiency [Belememis et al. 2020]. In order to mitigate this workload issue and explore modern multicore computing systems more properly, the VVC standard provides several native features to distribute the encoding effort among multiple processing units. These features can be employed independently or in combination to meet specific demands such as speed, quality, or bandwidth in video encoding. These features are commonly referred to as slices, tiles, and WPP.

2.1. Slices

Slices are data structures adopted in various video coding standards, with one of their primary objectives being resynchronization in the event of data loss during a video transmission, as well as enabling the transmission in data packets [Belememis et al. 2020]. In the VVC standard, slices can be categorized as raster-scan slices or rectangular slices. Raster-scan slices follow a sequential ordering of tiles, whereas rectangular slices cover a defined rectangular region of the picture. Unlike previous standards, VVC introduces greater flexibility in partitioning, allowing slices to exist within tiles and tiles to exist within slices. These structures play a crucial role in balancing parallel processing efficiency and coding performance [Wang et al. 2021]. Usually, slices can be classified as I, P and B types. I slices only support intraframe prediction, while P slices support both intra and inter. Besides, B supports bidirectional inter prediction [Wang et al. 2021].

2.2. Tiles

Tiles are another parallelism tool proposed in standards such as HEVC, OMedia Video 1 (AV1), and VVC. They consist of rectangular partitions created by defining horizontal and vertical boundaries within the image or frame [Chi et al. 2012]. These boundaries must ensure that whole CTUs are used, and splitting a CTU is not allowed. These structures can be either uniform or non-uniform in size. In uniform tiles, the encoder evenly distributes the number of CTUs within each partition, whereas in non-uniform partitions, the user can define the number of CTUs in each tile column (or row) according to their own criteria. However, encoders typically aim to establish a minimum number of samples per partition, which may vary depending on the encoder being used [Zhao et al. 2022]. Such structures are independent of each other, and this capability offers flexibility to the system, which can be leveraged to increase its parallelism potential, as independent segments can be processed by distinct cores, thus helping to reduce latency [Sullivan et al. 2012].

2.3. WPP

When the WPP tool is enabled, a slice is divided into CTU rows that can be processed in parallel. With WPP enabled, each available thread can process one CTU row. The first row is processed normally in raster scan order. For the second and subsequent rows, processing cannot begin until two CTUs from the previous row have been processed. This is due to the intra-frame prediction and motion vector information depending on data from the above and above-right CTUs. Another important aspect is that entropy coding also relies on previously encoded information from these CTUs [Papaioannou et al. 2020].

3. Overview of ERP 360-degree Video Coding

360-degree videos provide a spherical representation of a scene and are becoming popular in various fields, such as entertainment, education, and virtual tourism. Their popularity stems from their ability to deliver an immersive experience, allowing viewers to explore the environment interactively [Skupin et al. 2017]. Despite their spherical nature, there is no specific standard designed for encoding videos in this spherical domain. This presents a significant challenge, as the lack of a standard hinders interoperability between different platforms and encoding tools, creating the need for solutions that can handle this diversity.

To overcome this limitation, 360-degree videos are projected onto a rectangular surface, similar to world maps, allowing them to be encoded using conventional techniques. This projection is essential for adapting spherical content to formats that can be processed by existing video encoders, but it also introduces distortions that must be considered during the encoding process. Among the available projections, the Equirectangular Projection (ERP) is the most common [Ye et al. 2018]. However, this projection introduces texture distortions, especially at the top and bottom edges of the frame, where pixels are stretched horizontally. These distortions can impact the efficiency of the encoding process, as texture characteristics influence the behavior of encoding tools.

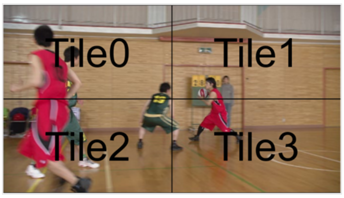
Additionally, 360-degree videos have a wider field of view than conventional videos, requiring significantly higher resolutions for accurate representation. This need for high resolution increases the amount of data to be processed and raises the complexity of the encoding process, making it more challenging and time-consuming. Therefore, it is important to evaluate the speedup potential and the coding efficiency impact of using the parallelism tools in VVC, such as tiles, for 360-degree videos. In this work, we present a set of evaluations to show the differences between 360-degree videos and conventional videos when the tile tool is used for VVC-based coding.

4. Evaluation environment

The experiments were carried out using the reference software encoder of the VVC standard, the VVC Test Model (VTM) version 20.2 [Bossen et al. 2024] along with the 360Lib version 13.1 [He et al. 2024]. All the experiments are performed according to the Common Test Conditions for 360-degree Videos [Hanhart et al. 2018], employing quantization parameters (QPs) 22, 27, 32, and 37. Furthermore, the All Intra configuration is used, in which only the intraframe prediction tools are enabled. Each video was encoded using one profile, five tile configurations, and four QP settings, resulting in a total of 20 tests per video. Due to the many experiments, only the first 100 video frames were encoded. The experiments were conducted on a server equipped with an Intel® Core™ i7-8700 CPU @ 3.20GHz $\times$ 12, 15.5GiB RAM and 64-bit Ubuntu 20.04 LTS operating system. To avoid oscillations in the workload, each experiment was run individually and in a fixed processing core, thus preventing the scheduling management from carrying out uncontrolled process migrations to other processing cores. This work is concerned with evaluating two main aspects of tile-based 360-degree video coding: workload distribution and coding efficiency. A series of conventional videos are also considered to compare the performance of 360-degree video encoding, and multiple tiling parameters are used to assess their suitability under different scenarios.

4.1. Tile subdivision

As discussed in Section 2.2, tiles are rectangular partitions composed of an integer number of CTUs. In the case of VVC, horizontal and vertical *boundaries* cross the complete frame width and height to determine the edges of tiles around the CTUs. Furthermore,



(a) Uniform tiles.

(b) Non-uniform tiles.

Figure 1. Example of tiles subdivision.

tiles can have uniform or non-uniform dimensions. When uniform tiling is employed, the boundaries are automatically placed at positions that evenly distribute the CTUs among available tiles, safeguarding the occurrence of an odd number of CTUs, where some tiles may have more or fewer CTUs than, as CTUs cannot be subdivided for this configuration. On the other hand, non-uniform tiling requires the user to define the positions of boundaries explicitly through the `TileColumnWidthArray` and `TileRowHeightArray` parameters. The boundaries are only placed at the edges of CTUs to guarantee an integer number of CTUs in each tile [Wang et al. 2021]. Figure 1 demonstrates one frame of the *BasketballDrive* video sequence partitioned into tiles following different configurations.

4.2. Video Sequences

For testing the 360-degree videos, their specific CTCs were used [Hanhart et al. 2018]. The evaluated sequences include the following videos: *ChairliftRide* (O1), *Gaslamp* (O2), *Harbor* (O3), *KiteFlite* (O4), *SkateboardInLot* (O5), *SkateboardTrick* (O6), *Train* (O7), and *Trolley* (O8). Even though the 360 videos are represented in 8k resolution (8192×4096), the CTCs require them to be downscaled to 4K (4432×2216) before encoding. Under these circumstances, each 360-degree video comprises 35×18 CTUs during encoding. For the tests conducted on conventional videos, six 4K (3840×2160) resolution videos from the CTCs of the VVC standard for conventional videos were used [Bossen et al. 2020]. The set of evaluated sequences includes: *Campfire* (C1), *CatRobot* (C2), *DaylightRoad2* (C3), *FoodMarket4* (C4), *ParkRunning3* (C5), and *Tango2* (C6). It should be noted that these conventional videos are composed of 30×17 CTUs.

4.3. Workload distribution

As one of the development stages of this study, video sequences were encoded without tiles (i.e., 1×1 tiles), with the entire frame being processed by a single processing core. The encoding times for each individual CTU were collected for the first 100 frames of the selected videos. Similar to [Storch et al. 2020], an algorithm was used to determine an optimal non-uniform tile subdivision to distribute the computational load as efficiently as possible among the available tiles. Since each tile can be processed by a separate processing core, this approach enhances the potential for parallelized system utilization. The main distinction of this study compared to the previously cited work lies in performing the tests with VVC standard while using 4K conventional and 360-degree videos.

The main novelty of this study is related to the inherently non-uniform distribution of texture complexity in 360-degree videos. Unlike conventional videos, where the computational load can be evenly distributed across any part of the frame, 360-degree videos exhibit a distinct pattern. The workload is more concentrated in the central region when analyzing the image vertically. Conversely, when the video frame is analyzed horizontally, even though the workload is more concentrated on the middle region, the left/right edges still represent a considerable encoding effort as will be demonstrated in Section 5. This behavior comes from the texture stretching caused by the ERP format.

4.4. Evaluated metrics

When parallel algorithms are considered, the most acceleration is obtained when the workload is perfectly balanced among processing cores. The speedup metric determines how much faster the VVC encoder can be when leveraging the parallelism enabled by the dependency break introduced by the tile tool compared to an encoding without tiles.

The speedup can be computed according to Equation (1). In this equation, $ReferenceTime$ represents the encoding time of a frame without using tiles, while $Max(TilesTime)$ corresponds to the encoding time of the slowest tile in the frame. This methodology is used because each tile can be processed by a different processing core, meaning that the total encoding time of the frame is directly determined by the tile that took the longest to complete its encoding process.

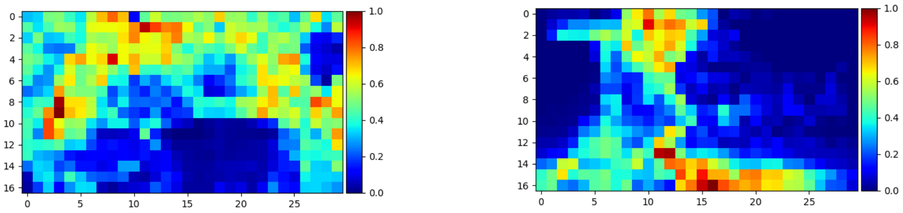
In this metric, if $Max(TilesTime)$ equals $ReferenceTime$, the result would be 1, indicating no speedup. Gains are only achieved when the result is greater than 1. For example, a speedup of 2 means the encoding process took half the time of $ReferenceTime$, a speedup of 3 means it took one-third of the time, and so on.

$$Speedup = \frac{ReferenceTime}{Max(TilesTime)} \tag{1}$$

Coding efficiency is measured using the Bjøntegaard Delta Rate (BD-Rate), which compares the difference in bitrate between two different encodings at the same visual quality. A positive BD-Rate value means that the evaluated encoding is less efficient than the baseline encoding, while a negative BD-Rate value indicates that it is more efficient. In other words, negative values signify an improvement in coding efficiency, whereas positive values indicate a decline in efficiency.

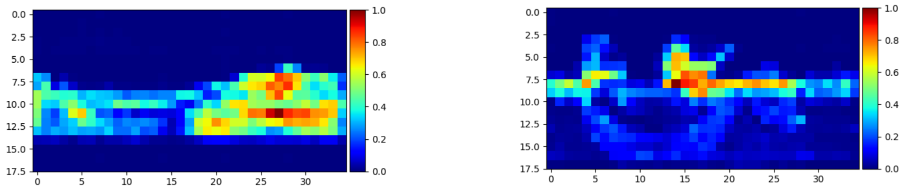
4.5. Best possible tiling encoding

Similarly to [Storch et al. 2019], this work is also concerned with identifying the maximum possible speedup during the encoding of tile 360-degree videos. To calculate the best tiling, the workload per CTU of the reference encoding without tiles was used. Based on this data, estimating the speedup for each non-uniform tiling was possible by rearranging hypothetical boundaries along the workload matrix extracted from the reference encoding. Therefore, the total workload within each hypothetical tile is extracted to obtain the limits that generate the best speedup. After this, a new tile pattern is created for each video sequence frame, and all sequences are encoded using these pre-computed non-uniform tiles. Based on this encoding, it is possible to evaluate the performance in terms of speedup increase and coding efficiency compared to an encoding with uniform tiles.



(a) Tango2 (C6). **(b) Campfire (C1).**

Figure 2. Spatial workload distribution for conventional videos .



(a) Harbor (O3). **(b) Train (O7).**

Figure 3. Spatial workload distribution for 360-degree videos.

5. Results

5.1. Spatial Workload Distribution

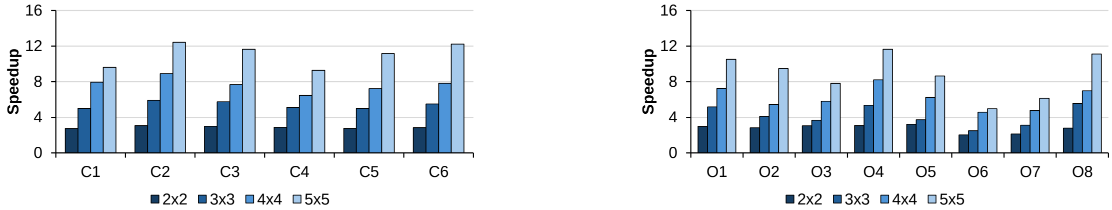
Figure 2 presents the normalized heatmap of CTU encoding times for a frame from the (a) *Tango2* (C6) and (b) *Campfire* (C1) conventional video sequences. These evaluations demonstrate that there is a significant workload imbalance across different regions of the frame. Similar to [Storch et al. 2020], an algorithm was used to determine an optimal non-uniform tile subdivision to distribute the computational load as efficiently as possible among the available tiles. Since each tile can be processed by a separate processing core, this approach enhances the potential for parallelized system utilization. The main distinctions of this study compared to the previously cited work include the fact that the current tests were conducted based on the VVC standard using 4K videos.

Figure 3 presents a similar spatial workload distribution analysis, but here, the 360-degree videos (a) *Harbor* (O3) and (b) *Train* (O7) are considered. Due to the way 360-degree video projection is designed and how their processing is implemented, encoding such videos reveals a concentration of higher computational load in the central region of the image, following a horizontal distribution.

5.2. Speedup for Videos with Uniform Tiles

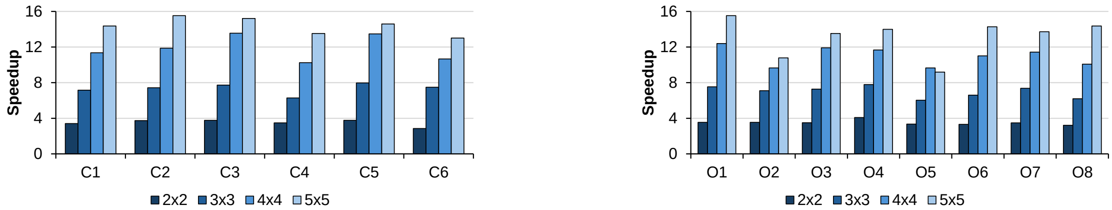
In this section, we present the speedup results obtained for both conventional and 360-degree videos using uniform tiles. The speedup was measured for four tile configurations, with the presented values representing the average across the four QP configurations used. Each result is shown in graphical form and discussed in the following sections.

In Figure 4a, the graph illustrates the speedup obtained for each of the four uniform tile configurations used (2×2, 3×3, 4×4, and 5×5) in conventional videos. A proportional increase in speedup is observed as the number of tiles increases, which is an expected behavior considering that, in this work, the number of tiles represents the number of cores working in parallel for processing. The lowest speedup recorded was 2.75 for the *Campfire* (C1) video with the 2×2 tile configuration, while the highest speedup achieved was 12.42 for the *CatRobot* (C2) video with the 5×5 tile configuration.



(a) Conventional videos. **(b) 360° videos.**

Figure 4. Speedup for uniform tiles.



(a) Conventional video. **(b) 360° video.**

Figure 5. Speedup for optimal tiles.

Figure 4b presents the speedup results in graphical format for 360-degree videos using uniform tile partitioning, with the lowest speedup value recorded at 2.02 (26% lower than the lowest value for conventional videos with uniform tiles) and the highest speedup value at 11.65 (6% lower than the highest value for conventional videos with uniform tiles). Considering all video sequences across all configurations, an average speedup of 5.47 is obtained for 360-degree videos, compared to an average of 6.75 for conventional videos. This average speedup, approximately 19% lower than that obtained for conventional videos, may indicate that workload imbalance is an even more critical issue for 360-degree videos to achieve their maximum speedup compared to conventional videos.

5.3. Speedup for Videos with Optimal Tiles

Significant speedup gains were observed in most tested sequences when the optimal tile division was used to better balance the workload for each video frame. These results are presented in Figure 5a, where it is visible that the speedup growth trend remains aligned with the increase in the number of tiles; however, the difference between the 4×4 and 5×5 tile configurations is less pronounced. In this test, the lowest speedup recorded was 2.85 for the *Tango2* (C6) video with the 2×2 tile configuration, while the highest speedup achieved was 15.52 for the *CatRobot* (C2) video with the 5×5 tile configuration.

Speedup differences between the optimal tiles and uniform tiles were observed. It can be seen that every single test conducted achieved a speedup gain, with the smallest gain being 0.01 for the video *Tango2* (C6) in the 2×2 configuration, while the highest gain was 6.25 for the video *ParkRunning3* (C5) in the 4×4 configuration. On average, when considering all configurations across these six videos, the speedup gain was 2.25.

When analyzing the graph in Figure 5b, which represents the speedup obtained by the optimal tiling configuration for 360-degree videos, it is possible to identify an instance where the previously observed pattern did not hold. In the *SkateboardInLot* (O5) video sequence with a 5×5 tile configuration, the speedup was lower than in the 4×4 tile configuration. Although the difference in speedup was only 0.47, this marks the first time in all conducted tests that a configuration with a higher number of tiles did not achieve a greater speedup. This suggests that, for the proposed method, depending on the spatial location of the highest workload within the frame, simply increasing the number of tiles does not always lead to better load balancing.

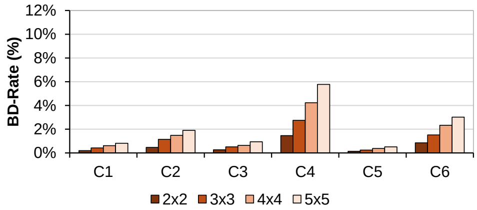
The lowest speedup recorded was 3.22 (13% higher than the lowest value for conventional videos with optimal tiles), while the highest speedup recorded was 15.52 (which, considering rounding, matches the highest value found for conventional videos with optimal tiles). Considering all video sequences across all configurations, the average speedup for 360-degree videos was 8.66, compared to an average of 9.27 in the tests conducted on conventional videos, once again, 360-degree videos achieve a lower average speedup compared to conventional videos, even when using optimal tiles.

When comparing the speedup obtained for 360-degree videos between optimal and uniform tiles, it is possible to observe gains in all cases, even for the *SkateboardInLot* (O5) video sequence with a 5×5 tile configuration, which, in Figure 5b, showed a lower speedup than the 4×4 configuration. In this case, the 5×5 configuration still provided a speedup gain, but to a lesser extent than the 4×4 configuration. A similar trend was observed in the*Gaslamp* (O2) and *KiteFlite* (O4) sequences, where the speedup gain in the 5×5 configuration was lower compared to the 4×4 configuration. The smallest speedup gain was 0.11 in the *SkateboardInLot* (O5) sequence with a 2×2 tile configuration, while the highest gain was 9.32 in the *SkateboardTrick* (O6) sequence with a 5×5 tile configuration. When the average speedup for all videos and configurations is considered, a mean speedup difference of 3.19 is found for 360-degree videos, compared to a mean difference of 2.52 for conventional videos. These findings indicate that the methodology tested in this study has greater potential for speedup gains in 360-degree videos than in conventional videos, with an observed gain 26.6% higher, according to the presented data.

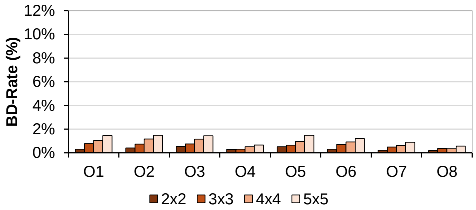
5.4. BD-Rate for Videos with Uniform Tiles

Beyond speedup, another aspect evaluated in this study is the variation in coding efficiency when using uniform and optimal tiles. During the tests, the bitrate and Peak Signal-to-Noise Ratio (PSNR) of all video sequences were extracted, and the BD-Rate was calculated. This evaluation is presented in Figure 6a, representing the coding efficiency of conventional video encoding using uniform tiles compared to encoding without tiles. Since positive BD-Rate values indicate a loss in coding efficiency, the graph shows that, regardless of the tested video sequence, coding efficiency tends to degrade as the number of tiles increases. This is expected, as the same dependency break that enables parallelism between tiles also eliminates the possibility of using references from adjacent tiles. Hence, the more tiles, the greater loss in coding efficiency is expected.

As observed with the coding efficiency data for conventional videos, the trend of increased degradation in this rate as the number of tiles increases remains valid for 360-degree videos – as presented in Figure 6b. In this test, there was only one exception, which occurred in the *Trolley* (O8) video sequence with a 4×4 tile configuration, where the BD-Rate was slightly lower (0.34%) compared to the 3×3 configuration (0.37%). Considering all the data presented in the graph of Figure 6b, an average BD-Rate of 0.73% was achieved for 360-degree videos, compared to an average BD-Rate of 1.36% for conventional videos, indicating less degradation in 360-degree videos compared to conventional videos in the tests conducted in this study.

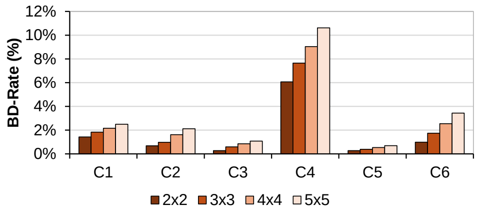


(a) Conventional videos.

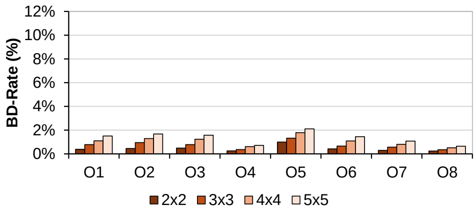


(b) 360° videos.

Figure 6. BD-Rate for uniform tiles.



(a) Conventional video.



(b) 360-degree video.

Figure 7. BD-Rate for optimal tiles.

5.5. BD-Rate for Videos with Optimal Tiles

Figure 7 presents the coding efficiency observed in the tests with the optimal tile configuration. Here, it is visible that using the optimal tiling leads to a larger coding efficiency penalty in most cases. The trend identified in the previous section remains the same, with a decline in coding efficiency as the number of tiles increases. As for the higher BD-Rate values compared to uniform tiles, this phenomenon likely happens because optimal tiles result in greater segmentation in areas with higher computational load, see Figure 3. These areas typically have a higher level of texture detail and motion, meaning that the dependency break in these regions has a worse impact on coding efficiency.

When analyzing the difference in BD-Rate obtained from an encoding based on optimal tiles compared to that with uniform tiles, all sequences degraded coding efficiency, except for the *CatRobot* (C2) sequence in the 3×3 tile configuration. Four out of the six 360-degree sequences (*CatRobot* (C2), *DaylightRoad2* (C3), *ParkRunning3* (C5), and *Tango2* (C6)) reached an average BD-Rate difference of 0.16%, which can be considered negligible. However, the results for the *Campfire* (C1) and *FoodMarket4* (C4) sequences cannot be overlooked, as they showed average BD-Rate increases of 1.4% and 4.79%, respectively, compared to uniform tiles. These findings highlight that while using optimal tiles configuration does not have a significant impact on coding efficiency for most videos, degradations of up to 178% may occur depending on their characteristics.

For 360-degree videos using optimal tiles, the average BD-Rate was 0.89%. In comparison, when conventional videos are considered, the average BD-Rate was 2.51%. Therefore, the tested technique has shown less degradation in 360-degree videos. Analyzing the BD-Rate difference obtained in the tests conducted with 360-degree videos using optimal versus uniform tiles, an average difference of 0.15% was found. When the same calculation was performed for conventional videos, the average difference was 1.15%. These results indicate that not only is the impact on coding efficiency lower for 360-degree videos compared to conventional ones but also that coding efficiency is less harmed when using optimal tiles for 360-degree videos compared to uniform tiles.

6. Conclusion

The results of this study highlight the effectiveness of using tiles in video encoding with the VVC codec, particularly in the context of 360-degree videos, which demonstrated greater efficiency compared to conventional videos. The analysis demonstrated that similar to conventional videos, applying uniform tiling during the encoding of 360-degree videos leads to a highly unbalanced workload distribution among processing cores. This, in turn, incurs the wastage of resources as some processing cores must stay idle while others are still conducting their work. Nonetheless, experimental evaluations showed more room for improvement in 360-degree videos when adaptive tiling strategies were considered. When conventional videos are considered, selecting the tiling pattern that produces the best speedup for each frame leads to a speedup increase of 87% compared to using uniform tiling, where the speedup of sequence C5 increased from 7.22 to 13.47. On the other hand, a similar experiment with 360-degree videos demonstrated that the speedup can be increased by 188% when adaptive tiling patterns are employed, where the speedup of sequence O6 increased from 4.96 to 14.28. Finally, this work also demonstrated that the coding efficiency degradation of tile-enabled video coding is less perceptible for 360-degree videos and that changing from a uniform tiling to the best possible tiling presents negligible coding efficiency variations when 360-degree videos are considered.

Acknowledgements

The authors thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) Finance Code 001, CNPq, FAPERGS and IFFar for funding this research.

References

Amestoy, T., Hamidouche, W., Bergeron, C., and Menard, D. (2020). Quality-driven dynamic vvc frame partitioning for efficient parallel processing. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3129–3133. IEEE.

Belememis, P., Panagou, N., Loukopoulos, T., and Koziri, M. (2020). Review and comparative analysis of parallel video encoding techniques for vvc. In *Applications of Digital Image Processing XLIII*, volume 11510, pages 258–276. SPIE.

Bossen, F., Boyce, J., Suehring, K., Li, X., and Seregin, V. (2020). Jvet-t2010: Vtm common test conditions and software reference configurations for sdr video. Technical report, JVET.

Bossen, F., Flynn, D., Li, X., Sharman, K., and Sühring, K. (2024). Vtm reference software for vvc. Accessed: Jan. 2024.

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.

Chi, C. C., Alvarez-Mesa, M., Juurlink, B., Clare, G., Henry, F., Pateux, S., and Schierl, T. (2012). Parallel scalability and efficiency of hevc parallelization approaches. *IEEE Transactions on circuits and systems for video technology*, 22(12):1827–1838.

Hanhart, P., Boyce, J., Choi, K., and Lin, J. (2018). Jvet-l1012: Jvet common test conditions and evaluation procedures for 360° video. Doc., JVET.

He, Y., Choi, K., Lin, J.-L., Sun, Y., Coban, M., Lu, Y., Abbas, A., Zhou, M., Deng, Z., and Oh, H.-M. (2024). Library for handling 360 degree panoramic projection formats. Accessed: Jan. 2024.

Jeong, J.-B., Lee, S., Kim, I., Lee, S., and Ryu, E.-S. (2020). Implementing vvc tile extractor for 360-degree video streaming using motion-constrained tile set. *Journal of Broadcast Engineering*, 25(7):1073–1080.

Lee, M., Song, H., Park, J., Jeon, B., Kang, J., Kim, J.-G., Lee, Y.-L., Kang, J.-W., and Sim, D. (2023). Overview of versatile video coding (h. 266/vvc) and its coding performance analysis. *IEIE Transactions on Smart Processing & Computing*, 12(2):122–154.

Margetis, G., Tsagkatakis, G., Stamou, S., and Stephanidis, C. (2023). Integrating visual and network data with deep learning for streaming video quality assessment.

Panagou, N., Belememis, P., and Koziri, M. (2022). Image segmentation methods for subpicture partitioning in the vvc video encoder. *Electronics*, 11(13):2070.

Papaioannou, G. I., Koziri, M. G., Papadopoulos, P. K., Loukopoulos, T., and Anagnostopoulos, I. (2020). Tile based wavefront parallelism in hevc. In *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA*, pages 1–5. IEEE.

Sandvine (2023). Global internet phenomena report. Accessed: Mar. 2025.

Skupin, R., Sanchez, Y., Wang, Y.-K., Hannuksela, M. M., Boyce, J., and Wien, M. (2017). Standardization status of 360 degree video coding and delivery. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE.

Storch, I., Agostini, L., Zatt, B., Bampi, S., and Palomino, D. (2021). Fastinter360: A fast inter mode decision for hevc 360 video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):3235–3249.

Storch, I., da Silva Cruz, L. A., Agostini, L., Zatt, B., and Palomino, D. (2019). The impacts of equirectangular 360-degrees videos in the intra-frame prediction of hevc. *Journal of Integrated Circuits and Systems*, 14(1):1–10.

Storch, I., Palomino, D., Zatt, B., and Agostini, L. (2020). Speedup evaluation of hevc parallel video coding using tiles. *J. of Real-Time Image Processing*, 17:1469–1486.

Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668.

Wang, Y.-K., Skupin, R., Hannuksela, M. M., Deshpande, S., Drugeon, V., Sjöberg, R., Choi, B., Seregin, V., Sanchez, Y., Boyce, J. M., et al. (2021). The high-level syntax of the versatile video coding (vvc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3779–3800.

Ye, Y., Alshina, E., and Boyce, J. (2018). Jvet-e1003: Algorithm descriptions of projection format conversion and video quality metrics in 360lib.

Zhao, S., Wenger, S., Sanchez, Y., Wang, Y.-K., and Hannuksela, M. M. (2022). RTP Payload Format for Versatile Video Coding (VVC). RFC 9328.