REAL-TIME UHD 4K HARDWARE DESIGN FOR THE GRADIENT-BASED SEARCH OF THE VVC AFFINE ME

Marcello M. Muñoz; Denis Maass; Murilo Perleberg; Luciano Agostini; Guilherme Corrêa; Marcelo Porto
Universidade Federal de Pelotas
{mmmunoz, dlmaass, mrperleberg, agostini, gcorrea, porto}@inf.ufpel.edu.br

1. INTRODUCTION

The streaming of digital videos is at an all-time high, with on-demand and live streaming accounting for over 68% of data transferred on the internet (Sandvine, 2024). As a consequence, continuous improvement in video compression efficiency is required. The state-of-the-art standard for Video Compression is the Versatile Video Coding (VVC) (Bross, 2020), which has the best coding efficiency among the current encoders.

The VVC standard incorporates multiple new tools and more advanced compression techniques than its predecessor, which enable VVC to achieve higher compression efficiency (Bross, 2021). In the inter-frame prediction, one of the new tools is the Affine Motion Estimation (AME), which is used to describe non-translation movements more accurately, such as scaling, skew, and rotation (Li. 2018).

The AME evaluates multiple sets of Motion Vectors (MVs), each of which results in a different predicted block. The AME is divided into two other algorithms: the Affine Merge Mode (MM), which utilizes the best motion vectors (MVs) from neighboring Coding Units (CUs), and the Affine Advanced Motion Vector Prediction (AMVP). The Affine MM has lower computational complexity since it only requires evaluating a small list of MVs. In contrast, the Affine AMVP has higher complexity, as it refines the starting MVs to find the MV with the minimal impact on encoding efficiency. Both algorithms require the Prediction Unit (PU) to be motion-compensated using the Affine mode with a precision of 1/16th of a pixel.

The Affine AMVP complexity stems from its use of two different algorithms to refine the MVs. The first is a Gradient-Based Iterative Algorithm (GBIA), and the second is a Block-Matching Iterative Algorithm (BMIA). The GBIA has fewer iterations but requires a more complex algorithm to minimize the error. BMIA searches the neighboring positions of the MVs for a PB with reduced error. The BMIA algorithm is relatively simple; however, the number of iterations is quite high. The AME is a high-complexity algorithm that leads to improved compression efficiency, but at a high computational cost. In this context, dedicated hardware design is needed to efficiently process the AME, especially for real-time Ultra-High Definition (UHD) video processing on battery-powered devices.

This paper presents a dedicated hardware design for the Affine Gradient-Based Search of the GBIA in the VVC Affine AMVP, capable of processing UHD 4K videos at 60 fps.

2. AFFINE MOTION ESTIMATION

The Affine Motion Estimation (AME) is one of the new tools added to the inter-prediction in the Versatile Video Coder (VVC). This tool better characterizes more complex motions such as scaling, rotation, skewing, and the combination of these types of motion (Bross, 2021). The AME searches the previously encoded frames for a similar block, similar to the original ME. VVC supports two Affine models: 4-parameters and 6-parameters, which utilize two or three MVs, respectively. The 4-parameter model uses two Corner Point Motion Vectors (CPMVs), located in the top-left and top-right of the Prediction Units (PU), which can be seen in Fig. 1 (a), while the 6-parameter model uses these two CPMVs but also has a bottom-left MV as well, which can be seen in Fig. 1 (b) (Li, 2018). The process of generating the predicted block is Affine Motion Compensation (AMC), which is processed in 4x4 subblocks, as shown in Fig. 1 (c). It utilizes filters selected using the Subblock Motion Vector (SMV). The VVC AME is divided into two different algorithms: the Affine Merge Mode (MM) and the Affine Advanced Motion Vector Prediction (AMVP). Affine MM utilizes the best motion vectors (MVs) encountered in the neighboring CU, which reduces the number of iterations required to process the PU. The Affine AMVP requires a high number of iterations because it uses a Gradient-Based Iterative Algorithm (GBIA) and a block-matching iterative algorithm to refine CPMV.

Both algorithms begin by applying the AMC with starting CPMVs to generate a predicted block. In GBIA, Sobel filters (X and Y directions) are applied to compute affine coefficients, which are used to form a matrix (4×4 or 6×6, depending on the model). An error column—based on the difference between predicted and original samples—is added, creating a system of linear equations (SLE). Solving the SLE provides four or six parameters, which are then used to update motion vectors (MVs). The process iterates up to five times or until the motion vector change becomes zero, indicating the optimal CPMV is found.

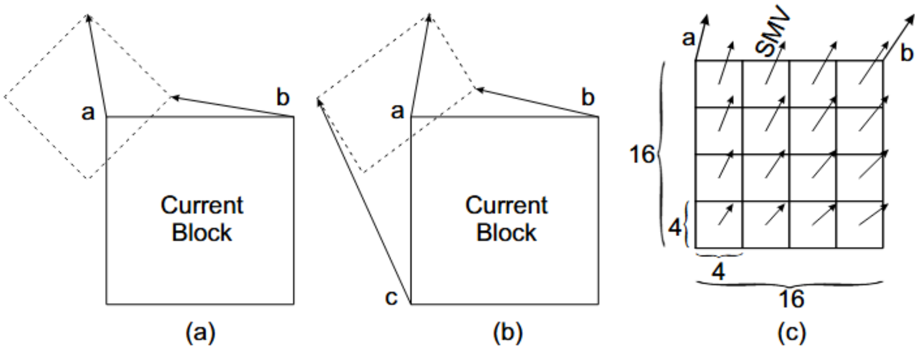

Fig. 1. Affine models: (a) 4-Parameter model with LT and RT MVs; (b) 6-Parameter model with LT, RT, and LB MVs. (c) Affine representation of the subblock split, where 16 subblocks with 4x4 samples compose one 16x16 CU. Each subblock has its own SMV.

3. AFFINE GRADIENT-BASED SEARCH ARCHITECTURE

This section presents the architecture for the Affine Gradient-Based Search (GBIA). The design refines motion vectors (MVs) by combining AMC-generated predictions with error feedback, producing new candidate prediction motion vectors (CPMVs). Each cycle processes one row of 16 predicted and original samples, generating CPMVs depending on the affine model.

The architecture consists of several key components:

- **Row Buffer, Sobel Engine, Error Generator** – prepare gradients and residuals.

- **Parameter Generator** – forms affine coefficients and system of linear equations (SLEs), aggregated via an Adder Tree and adapted for four-parameter cases.
- **System Solver** – converts SLEs to floating-point form, solves with Gauss–Jordan elimination, and produces CPMVs in the Delta MV stage.

Pipelining enables simultaneous SLE generation and solving, supporting two PUs concurrently. SLE generation is cycle-intensive, while solving takes 8–12 cycles depending on the model.

Performance tests with VVC (VTM 16.2) on UHD 4K sequences showed a modest BD-Rate increase (~0.92% overall), with Class A2 requiring the highest load. The calculated operational frequency is ~147 MHz to sustain 60 fps at 4K.

4. SYNTHESIS RESULTS AND RELATED WORKS

This section reports the ASIC synthesis results of the proposed GBIA architecture, implemented in TSMC 40nm standard-cell technology using Cadence RTL Compiler. The synthesis target frequency was set to 147 MHz, corresponding to the throughput required for UHD 4K videos at 60 fps, as derived from the CTC A2 class sequences.

Table I summarizes the area (in equivalent NAND2 gates) and power dissipation results for the proposed design, measured under realistic switching activity from the first 1.5K PUs of all six UHD 4K CTC sequences. The reported power values represent the average across all sequences.

Comparative analysis highlights the limited number of hardware-oriented studies on Affine AMVP. Taranto, 2022 and Chen, 2024 both introduce simplifications, such as eliminating iterative search, reducing interpolation, and limiting PU sizes, which lower implementation complexity but lead to BD-rate increases (unreported in Taranto, 2022 and up to 1.835% in Chen, 2024). Hong, 2025 instead modifies the Affine AMVP by merging GBIA and BMIA, extending iteration counts, and enabling 8×8 PUs. While this reduces BD-rate penalties (0.315–0.332%), its frequency estimation—based on resolution rather than actual PU counts—underestimates computational load.

In contrast, the proposed architecture retains up to three iterations of the Gradient-Based Search, unlike Taranto, 2022 and Chen, 2024, thereby preserving coding efficiency and achieving BD-rate gains at the cost of increased computational complexity. Relative to Hong, 2025, the proposed design avoids the scalability issues associated with additional PU sizes while using realistic input-driven frequency estimation for accurate performance evaluation.

TABLE I
COMPARISON BETWEEN THE PROPOSED ARCHITECTURE AND RELATED WORKS.

	Hong [18]	Chen [19]	Taranto [17]	Proposed
Design	AME	AME	AME	Gradient-Based Search
Technology	Virtex 7	28nm	45nm	40nm
Area (Gates)	-	1313K	29.9K	728K
Frequency	200MHz	500MHz	341MHz	147MHz
Power	230.50mW	156.83mW	1.82mW	54.70mW
BD-Rate	-	0.492%∼1.835%	not presented	0.92%
Resolution FPS	UHD 4K @60	UHD 4K @6.8∼120	UHD 4K @60	UHD 4K @60

5. CONCLUSION

This work presents a hardware design for the gradient-based search module of the VVC Affine Motion Estimation (AME), capable of processing UHD 4K video at 60 frames per second. The proposed architecture supports 16×16 PUs, handles both 4-parameter and 6-parameter affine models, and generates the next set of Candidate Predictive Motion Vectors (CPMVs) to be evaluated in the Gradient-Based Iterative Algorithm (GBIA). The design employs a pipelined structure that enables the simultaneous processing of two PUs, allowing it to sustain the average number of predicted PUs required by AME with up to three gradient search iterations, contributing to improved coding efficiency. The architecture was synthesized using TSMC 40nm standard-cell technology. Synthesis results show that, for UHD 4K video at 60fps, the design achieves an area of 728K Gates, dissipates 18.86mW of power at 147 MHz, and incurs a BD-Rate increase of only 0.92%.

5. BIBLIOGRAPHY

Sandvine, “2024 global internet phenomena report,”https://www.sandvine.com/global-internet-phenomena-report-2024”, 2024, Retrieved July 26, 2024.

B. Bross, J. Chen, S. Liu, and Y.-K. Wang, “Versatile Video Coding Editorial Refinements on Draft 10,” JVET-T2001, October 2020.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J.Sullivan, and Jens-Rainer Ohm, “Overview of the Versatile Video Coding (VVC) Standard and its Applications,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 10, pp. 3736–3764, 2021.

Li Li, Houqiang Li, Dong Liu, Zhu Li, Haitao Yang, Sixin Lin, Huanbang Chen, and Feng Wu, “An efficient four-parameter affine motion model for video coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 8, pp. 1934–1948, 2018.

Costantino Taranto, Simplified affine motion estimation algorithm and architecture for the versatile video coding standard, Ph.D. thesis, Politecnico di Torino, 2022. Jingping Hong, Zhihong Dong, Mengxin Pang, Zetao Kang, and Peng

Cao, “Hardware implementation of iterative method for enhanced affine motion estimation in versatile video coding,” Journal of Real-Time Image Processing, vol. 22, no. 1, pp. 1–13, 2025.

Shushi Chen, Leilei Huang, Zhao Zan, Zhijian Hao, Hao Zhang, Xiaoxiang Chen, Minge Jing, Xiaoyang Zeng, and Yibo Fan, “Affine motion estimation hardware implementation with 51.7%/67.5% internal bandwidth reduction for versatile video coding,” IEEE Transactions on Circuits and Systems for Video Technology, 2024.