Uniformly Accelerated Motion Model for Inter Prediction

Zhuoyuan Li, Yao Li, Chuanbo Tang, Li Li, Dong Liu, Feng Wu*

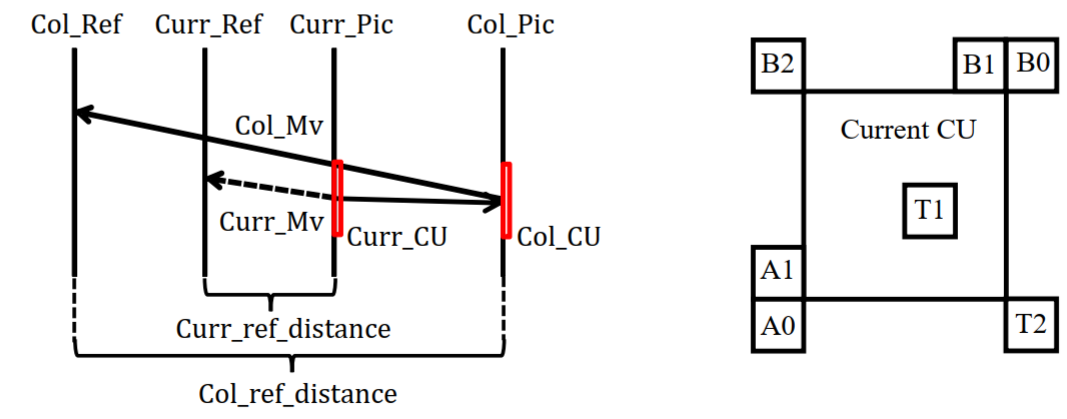
University of Science and Technology of China (USTC), Hefei, Anhui 230027, China

{zhuoyuanli, mrliyao, cbtang}@mail.ustc.edu.cn, {lil1, dongeliu, fengwu}@ustc.edu.cn

1. Contributions

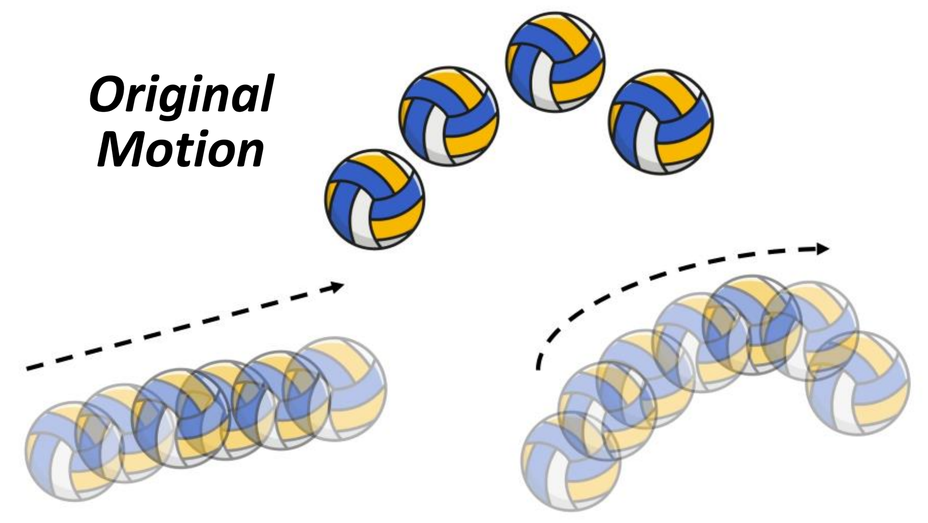
1. To the best of our knowledge, we are the first to introduce the ***uniformly accelerated motion model*** (UAMM) to improve the performance of inter prediction.
2. To adapt to the video coding framework, we formulate the usage of UAMM, and ***propose the UAMM-based parameter derivation and extrapolation schemes in the predictive process***.
3. We integrate the uniformly accelerated temporal motion model into existing inter modes (Merge, MMVD, CIIP) of VVC to ***achieve higher prediction accuracy***.

2. Concept

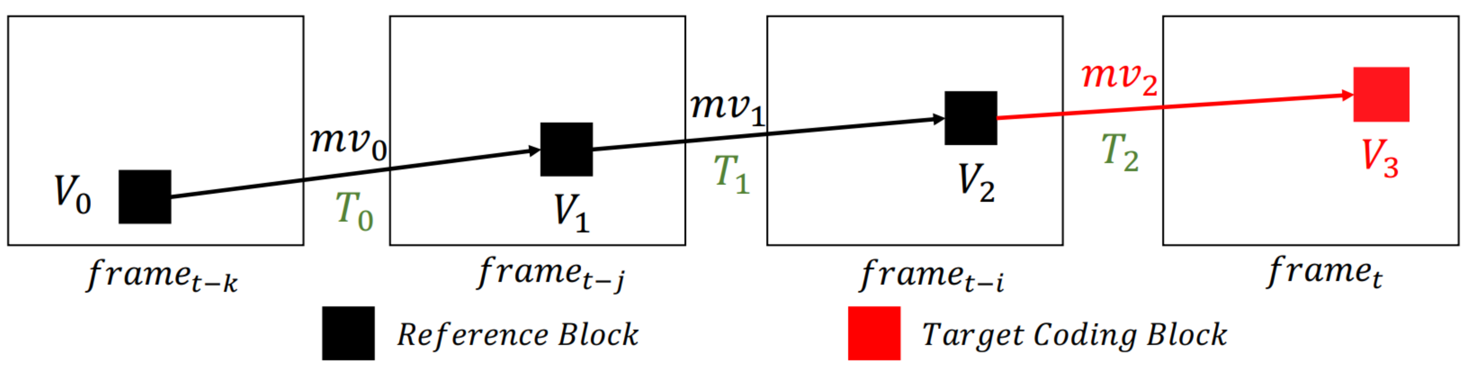


$$Curr\_Mv = \frac{Curr\_ref\_distance}{Col\_ref\_distance} \times Col\_Mv$$

Temporal motion vector prediction (TMVP) in VVC



3. Motion Modeling



Motion trajectory between the coding (to-be-coded) block and reference (reconstructed) blocks in previous reference frames.

Uniformly Accelerated Model

$$x_t = v_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2,$$

$$v(t) = v_0 + a \cdot t,$$

Integrated into Coding Process

$$mv_0 = v_0 \cdot T_0 + \frac{1}{2} \cdot a \cdot T_0^2,$$

$$mv_1 = v_1 \cdot T_1 + \frac{1}{2} \cdot a \cdot T_1^2,$$

$$v_1 = v_0 + a \cdot T_0,$$

Accelerated Model Parameter Derivation

$$a = \frac{2\,(mv_1 \cdot T_0 - mv_0 \cdot T_1)}{T_0 T_1 (T_0 + T_1)},$$

$$v_0 = \frac{mv_0 - \frac{a \cdot T_0^2}{2}}{T_0}.$$

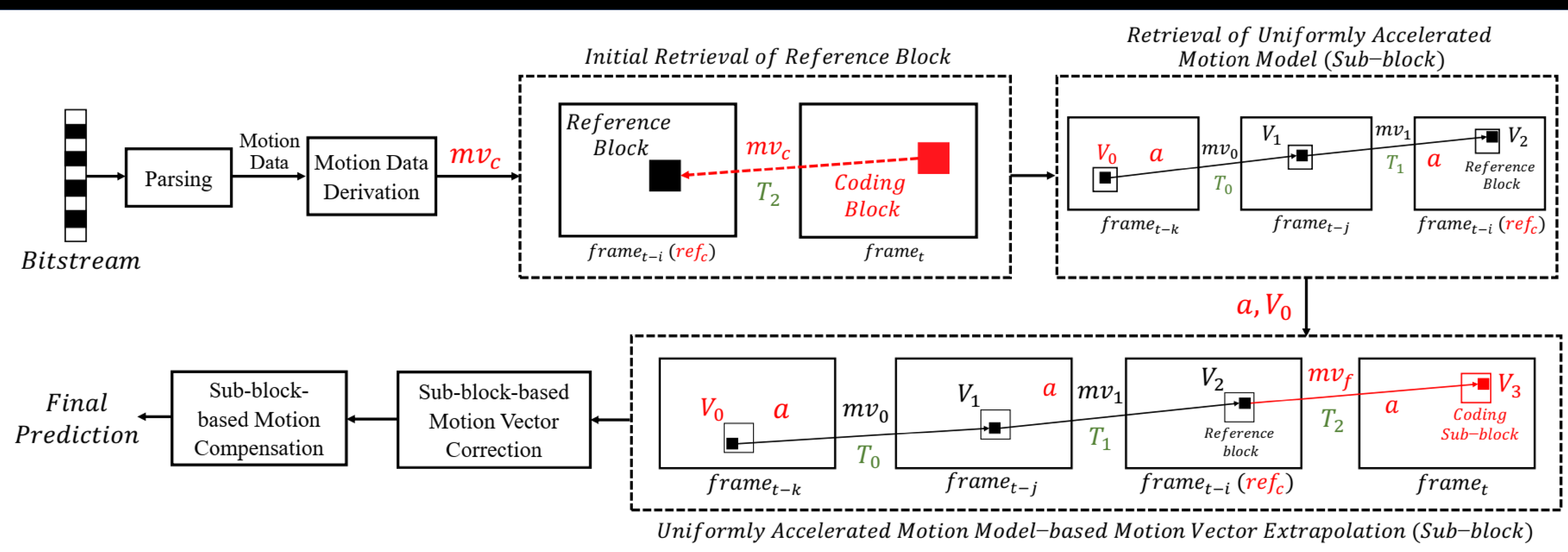
4. Framework

Accelerated Model Parameter Extrapolation

$$mv_2 = v_2 \cdot T_2 + \frac{1}{2} \cdot a \cdot T_2^2,$$

$$mv_2 = v_0 \cdot T_2 + a \cdot T_2 (T_0 + T_1) + \frac{1}{2} \cdot a \cdot T_2^2.$$

Note that ***the parameters are all represented in vector form***, and both the horizontal component and the vertical component can be derived simultaneously.



Decoding Process of UAMM in VVC

*The proposed **uniformly accelerated motion model-integrated inter prediction framework***

5. Experiment Results

Coding Performance and Relative Complexity Based on VTM-12.0 under LDP Configuration

Class	Sequence Name	Low-delay P Y	Low-delay P U	Low-delay P V
ClassB (1920x1080)	*MarketPlace*	-0.11%	0.43%	0.47%
	RitualDance	0.01%	0.10%	-0.04%
	Cactus	-0.10%	-0.27%	-0.02%
	BasketballDrive	-0.12%	0.05%	-0.03%
	BQTerrace	0.03%	0.01%	-0.04%
ClassC (832x480)	*BasketballDrill*	-0.16%	0.22%	0.02%
	BQMall	-0.02%	-0.09%	-0.20%
	PartyScene	-0.05%	-0.28%	0.24%
	RaceHorsesC	0.06%	-0.46%	0.70%
ClassD (416x240)	*BasketballPass*	-0.16%	0.44%	-0.85%
	BQSquare	-0.14%	-0.54%	-0.53%
	BlowingBubbles	-0.15%	-0.82%	-0.12%
	RaceHorses	-0.28%	0.34%	-1.02%
ClassE (1280x720)	*FourPeople*	-0.35%	-0.49%	-0.43%
	Johnny	-0.38%	0.29%	-1.23%
	KristenAndSara	-0.08%	0.27%	0.14%
Overall		**-0.13%**	**-0.05%**	**-0.17%**
EncT			101%	
DecT			102%	

BD-Rate Results (Y Component) Compared to VTM-12.0 on Specific Scenes

Sequence	Resolution	LDP (%)
Beauty	*1080P*	-0.16%
Jockey	*1080P*	-0.19%
WestLibrary	*1080P*	-0.15%
WestLabBuilding	*1080P*	-0.21%
CrossRoad	*1080P*	-0.14%
CrossRoad2	*1080P*	-0.12%
Overall		**-0.16%**
EncT		101%
DecT		103%

BD-Rate Results of Some Sequences Based on VTM-12.0 under LDP Configuration

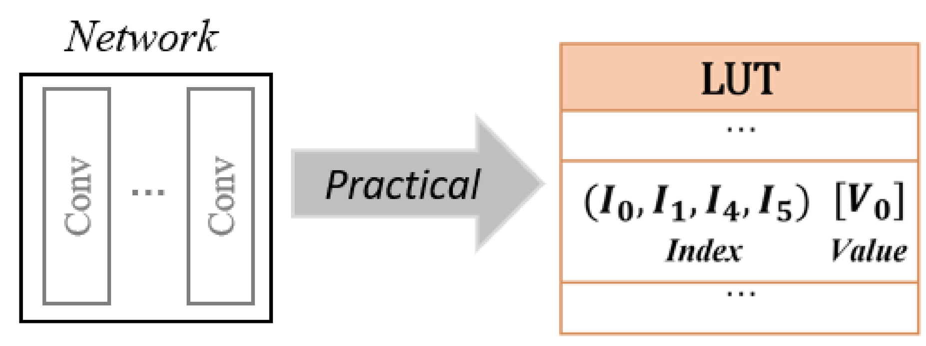
Sequence	*BQMall*	*BQSquare*	*Johnny*	*BlowingBubbles*	*BasketballPass*
BD-rate	-0.19%	-0.35%	-0.11%	-0.11%	-0.15%

Conclusion: UAMM can efficiently deal with some complex scenarios that ***TMVP and SbtMvp of VVC*** are difficult to address.

In-Loop Filtering via Trained Look-Up Tables

(Oral Session 9, 15:30–17:00)

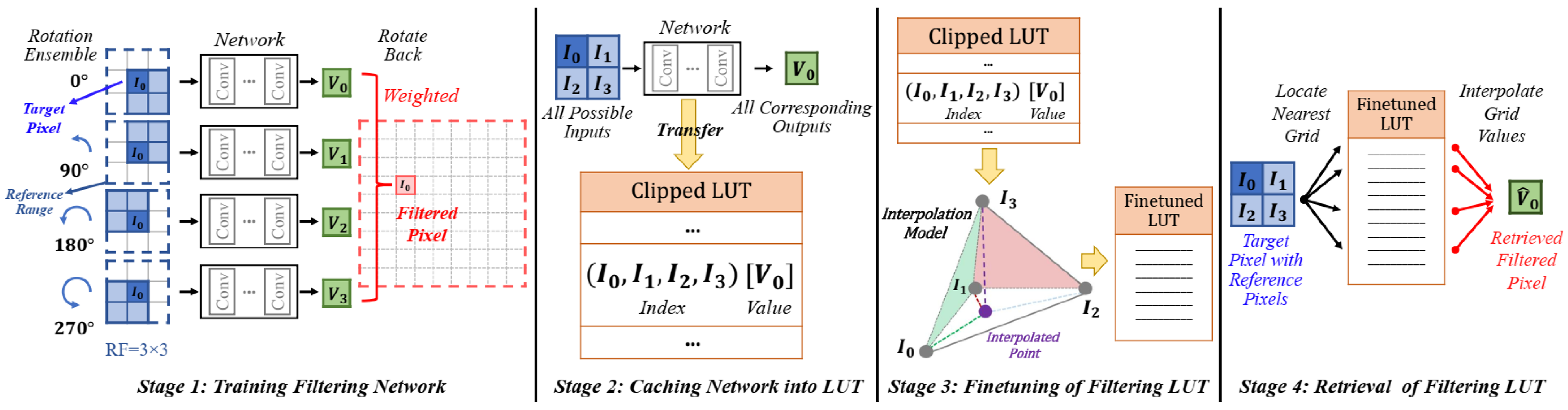
Zhuoyuan Li, Jiacheng Li, Yao Li, Li Li, Dong Liu*, Feng Wu



Contribution: 1. We study an efficient in-loop filtering scheme by adopting look-up tables (LUTs). 2. Our method has ***explored a new and more practical approach for neural network-based Coding Tools***.

Basic idea: Adopting the look-up operation of LUT to replace the inference process of deep neural network (DNN) in coding process, which is friendly for embedded systems to accelerate computation with far fewer floating-point operations.

Scheme: To achieve this goal, we establish a LUT-based in-loop filtering framework (***LUT-ILF***), and introduce a series of LUT-related modules to strengthen its efficiency, including the enhancement of filtering reference range with the limited LUT size (***progressive indexing and reference indexing***), the optimization of LUT size with limited memory cost (***clipping / finetuning***), the selection of reference pixels (***learnable weighting***).



*Basic framework of **look-up table-based in-loop filtering framework** (**LUT-ILF**)*

Experiment Results: Compared to the low/high complexity operation point setting (LOP, HOP) of ***NNVC-ILF***, our ultrafast mode (***LUT-ILF-U***, reference range: 5×5, **0.13 kMAC/pixel**, 164 KB), very fast mode (***LUT-ILF-V***, reference range: 9×9, **0.40 kMAC/pixel**, 492 KB) and fast mode (***LUT-ILF-F***, reference range: 13×13, **0.93 kMAC/pixel**, 1148 KB) ***provide a series of new trade-off points that show lower time and computational complexity and good performance beyond VVC.***

Methods	BD-Rate (AI)	BD-Rate (RA)	Computational Complexity	Storage Cost	Energy Cost[2]	Time Complexity (enc/dec, CPU)
NNVC-LOP[1] (VTM-11.0)	-4.61%~-4.78%	-5.20%~-5.37%	17.0 kMACs/pixel	129.98 KB (*int16*) 228.33 KB (*float*)	11900 *pJ* (*int16*) 78200 *pJ* (*float*)	108%/4717%~109%/4724% (AI) 114%/8274%~114%/8322% (RA)
NNVC-HOP[1] (VTM-11.0)	-7.79%~-7.91%	-10.12%~-10.31%	477.0 kMACs/pixel	2826.2 KB (*int16*) 7444.5 KB (*float*)	333900 *pJ* (*int16*) 2194200 *pJ* (*float*)	133%/24372%~276%/134057% (AI) 159%/43509%~399%/227720% (RA)
***LUT-ILF-U* (VTM-11.0)**	**-0.13%**	**-0.10%**	**0.13 kMACs/pixel**	**164 KB (*int8*)[3]**	**180.2 *pJ***	**101%/102% (AI), 101%/105% (RA)**
***LUT-ILF-V* (VTM-11.0)**	**-0.34%**	**-0.27%**	**0.40 kMACs/pixel**	**492 KB (*int8*)**	**497.2 *pJ***	**102%/103% (AI), 103%/106% (RA)**
***LUT-ILF-F* (VTM-11.0)**	**-0.51%**	**-0.39%**	**0.93 kMACs/pixel**	**1148 KB (*int8*)**	**1163.25 *pJ***	**102%/106% (AI), 104%/108% (RA)**

Contact: zhuoyuanli@mail.ustc.edu.cn

Homepage: https://zhuoyuanli1997.github.io/

WeChat QQ LinkedIn