Fast Intra Mode Decision and Block Matching for HEVC Screen Content Compression

Hao Zhang, Qiao Zhou ,NingNing Shi ,Feng Yang, Xin Feng, Zhan Ma
hao@csu.edu.cn , mazhan@nju.edu.cn
Central South University, China; *Nanjing University*, China; *Chongqing University of Technology,* China

ABSTRACT

• A fast intra mode decision scheme for HEVC screen content coding is proposed to reduce the complexity of intra mode decision search for each coding unit.

• A fast block matching scheme for intra block copy is proposed to reduce the number of blocks for 2-D search.

• 39% and 35% encoding time reduction for lossy and lossless encoding scenarios with negligible quality loss under the SCC common test condition.

SCREEN CONTENT VIDEO

• Screen content video, often containing mixed content consisting of natural video, text, graphics, image, video, etc., has been widely embedded in various applications (wireless displays, shared screen collaboration, virtual desktop interface etc.) in recent years.

• Screen content is different from the natural video and it is inevitable to develop the new tools to efficiently exploit the correlation in screen images exemplified in Figure1.

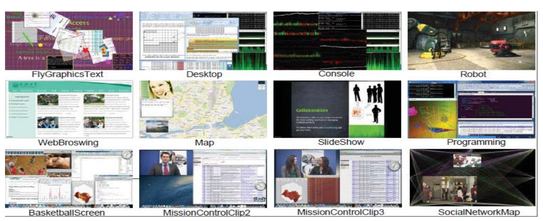

Fig. 1: Illustration of typical screen content image in daily practice.

PROPOSED ALGORITHM

Fast Intra Mode Decision Based on Background Detection

• Our contribution here is to speed up the mode decision process if a CU block is estimated as stationary, where its partition is closely related to the collocated CU.
• Firstly, exploit temporal correlation to speed up intra prediction, and compare the depth of current CU and the depth of collocated CU in previous encoded frame.
• Then, apply the background detection to judge whether the CU is stationary or not.
• Finally, to avoid the accumulation of prediction error, we switch off the fast algorithm every ten frames.

Fast Block Matching Based on Adaptive Searching Step Size Adjustment

• A method of adjusting the step sizes for the 2D search process is proposed.
• The search position x = x + dx, where x represents the horizontal search position and dx denotes the original searching step size. The original value of dx is set to one or two, depending on the search positions.
• The main idea of the proposed fast search is to dynamically adjust dx to dx' by the rules detailed in the following:
In the case of dx = 1 :
If SAD > 2SAD$_0$, dx' = 2.
If SAD > 4SAD$_0$, dx' = 4.
In the case of dx = 2 :
If SAD > 2SAD$_0$, dx' = 4.
If SAD > 4SAD$_0$, dx' = 6.
• Here SAD$_0$ represents the Mth smallest SAD value at the end of the SAD queue.

RESULTS

• This section presents the experimental results with our proposed solution.
• The test sequences are the screen content representing popular and typical screen content application scenarios consist of four categories sequences.
• Results are shown in Table 1 and Table 2 with BD-Rate performance and encoder time reduction for AI lossless and lossy encoding.

Table 1: Performance Evaluation for All Intra Lossy coding of the proposed fast mode decision against SCM 3.0

All Intra				
		BD-Rate Increase		
		G-Y	B-U	R-V
Lossy	TGM-G	1.3%	1.5%	1.5%
	MIX-G	1.0%	1.1%	1.2%
	AMT-G	0.1%	0.2%	0.2%
	CAC-G	0.0%	0.0%	0.0%
	TGM-Y	1.4%	1.4%	1.4%
	MIX-Y	1.1%	1.1%	1.1%
	AMT-Y	0.3%	0.4%	0.3%
	CAC-Y	0.0%	0.0%	0.0%
	Overall	0.7%	0.7%	0.7%
	Enc Time[%]	61%		

Table 2: Performance Evaluation for All Intra Lossless coding of the proposed fast mode decision against SCM 3.0

All Intra					
		Bit-Rate Increase			
		Total	Average	Min	Max
Lossless	TGM-G	0.7%	0.9%	0.1%	2.9%
	MIX-G	0.4%	0.4%	0.2%	0.5%
	AMT-G	0.0%	0.0%	0.0%	0.0%
	CAC-G	0.0%	0.0%	0.0%	0.0%
	TGM-Y	0.4%	0.4%	0.1%	0.8%
	MIX-Y	0.1%	0.1%	0.1%	0.1%
	AMT-Y	0.0%	0.0%	0.0%	0.0%
	CAC-Y	0.0%	0.0%	0.0%	0.0%
	Overall	0.2%	0.2%	0.1%	0.5%
	Enc Time[%]	65%			

CONCLUSIONS

We propose two novel fast algorithms for screen content encoding in this paper. Compared with the reference software SCM 3.0, the proposed algorithms provide the averaged 39% and 35% encoding time reduction with only 0.7% BD-Rate increase and 0.2% bit-rate increase for AI lossy and lossless cases, respectively.