Joint Collaborative Team on Video Coding (JCT-VC)
of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11
2nd Meeting: Geneva, CH, 21-28 July, 2010

Document: JCTVC-B089

Title:	**TE2: Compressed Reference Frame Buffers (CRFB)**
Status:	Input Document to JCT-VC
Purpose:	Proposal

Author(s) or Contact(s):	Mehmet Umut Demircin Texas Instruments Inc., USA	Tel: Email:	+1-214-567-4005 demircin@ti.com
	Madhukar Budagavi Texas Instruments Inc., USA	Tel: Email:	+1-214-480-1861 madhukar@ti.com
	Minhua Zhou Texas Instruments Inc., USA	Tel: Email:	+1-214-480-3816 zhou@ti.com
	Salih Dikbas Texas Instruments Inc., USA	Tel: Email:	+1-214-567-6650 salih@ti.com
Source:	Texas Instruments Inc.		

Abstract

An in-loop memory access bandwidth reduction technique is proposed. Proposed tool compresses the reference pictures before they are written to the memory and decompresses before they are being read. Fixed compression ratio is targeted for a block of pixels to enable random access. Proposed technique provides 12 bit/pixel to 8 bit/pixel compression when Internal Bit-Depth Increase (IBDI) tool is turned on. 8 bit/pixel to 4 bit/pixel compression is achieved when IBDI is disabled. 2-D integer S-transform on 8x8 blocks, DC prediction, quantization and variable length entropy coding is employed. Performance is tested and verified as a part of the Tool Experiment 2 (TE2). Proposed algorithm results a worst case BD-Rate increase of 1.01% for Class A, B and E bitstreams for IBDI-on and 0.80% for IBDI-off settings.

1 Introduction

The purpose of this algorithm is to reduce external memory access bandwidth and the reference frame buffer memory size in video coding. The reference frames are compressed before being stored in memory and they are decompressed after being read from the memory. Figure 1 illustrates where the compressed reference frame buffer (CRFB) algorithm is executed within the simplified video encoding block diagram. Decoder implementation is similar. An initial version of this algorithm was proposed in [1] and [2]. Two key features of the proposed technique are: (1) Fixed compression ratio for maintaining random access to predefined blocks of pixels in the memory. (2) Carrying out reference frame compression in the core video coding loop so that quantization errors show up in the residual after motion compensation thereby preventing drift between the encoder and the decoder. Proposed algorithm operates on the raw YUV format (YUV420, 8-bit (IBDI off) or 12-bit (IBDI on)) frames. CRFB algorithm has two modes, IBDI-on and IBDI-off. 12 bit/pixel to 8 bit/pixel compression is performed when Internal Bit-Depth Increase (IBDI) tool is turned on. Basic memory access unit is in 8x8 blocks in the IBDI-on mode. 8 bit/pixel to 4 bit/pixel compression is achieved when IBDI is disabled. Memory can be accessed randomly in row of MBs in the IBDI-off mode.

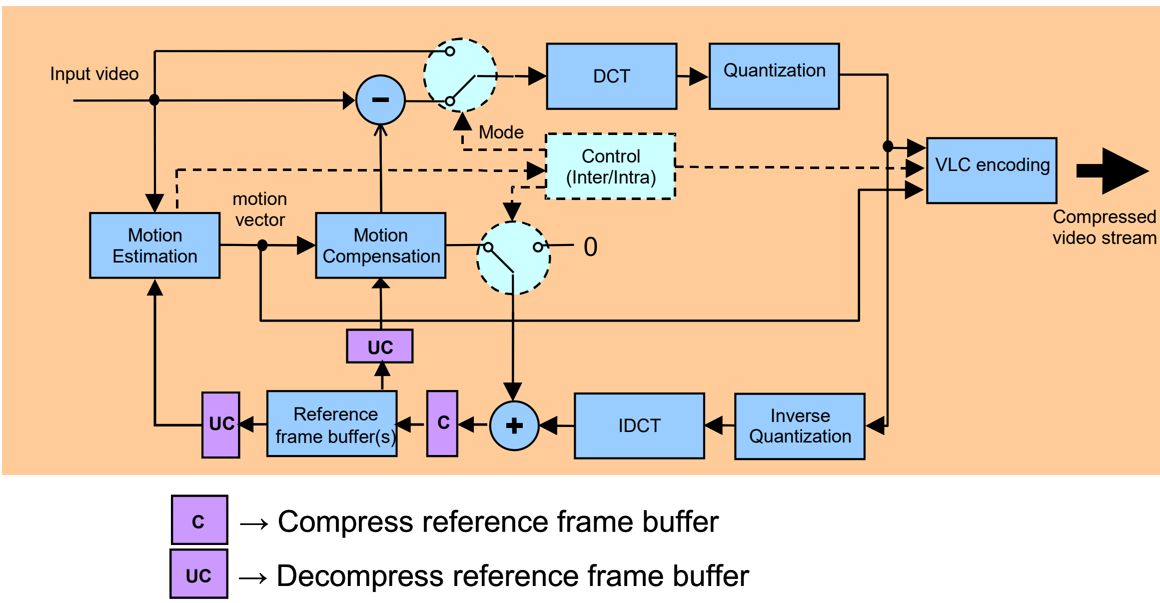

Figure 1- Video coding using compressed reference frame buffers (CRFB).

2 Algorithm Description

Reference frame compression algorithm is designed based on the following goals:

- Random memory access to predefined pixel blocks.
- Low-complexity implementation.
- Minimal quality degradation.

Compressed reference frame buffer algorithm consists of basic image compression tools as depicted in Figure 2. Transformation, quantization, DC prediction (for IBDI-off mode) and entropy coding are designed to minimize the hardware and software implementation complexity. The details of these techniques and proposed CRFB encoder are discussed in the following sections.

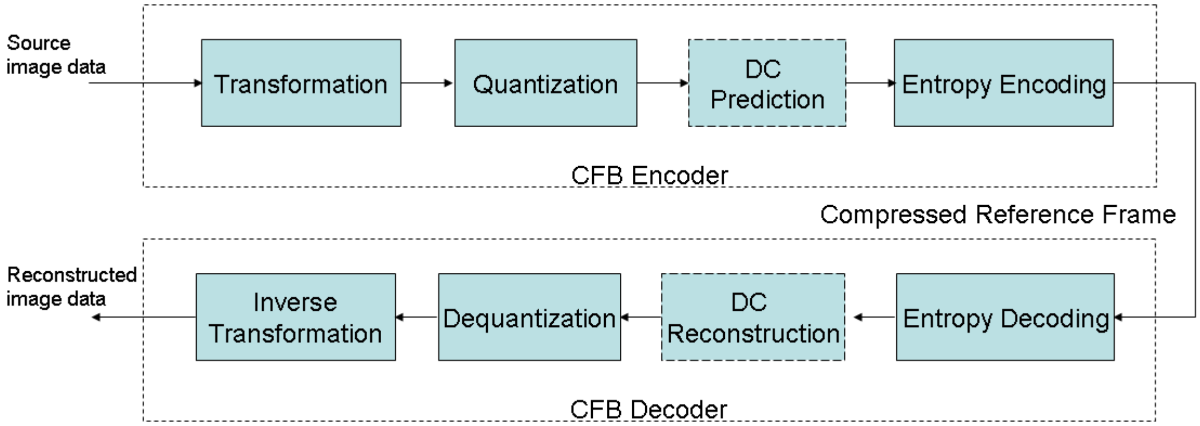

Figure 2 – Compressed reference frame buffer (CRFB) algorithm block diagram.

2.1 CRFB Transformation Algorithm

2-D S transform, which is an integer transform based on lifting scheme, is selected as the transform [3], [4]. Transform operates on 8x8 Y, 4x4 U, and 4x4 V pixel blocks, which will be referred as CRFB block here on. 2-D S transform is separable and applied in horizontal and vertical directions consecutively. The order is not important, however, the forward and inverse transforms should have reverse orders of each other.

S transform has several advantages: 1) enables direct fixed-point implementation, 2) enables lossless coding, and 3) enables low-complexity implementation. S transform is reversible and implementable with only addition, subtraction and shift operations.

Illustration of forward one lifting step is shown in Figure 3, where P denotes prediction and U denotes the update stages. Inverse transform is obtained by reversing the steps of the forward transform and flipping the signs.

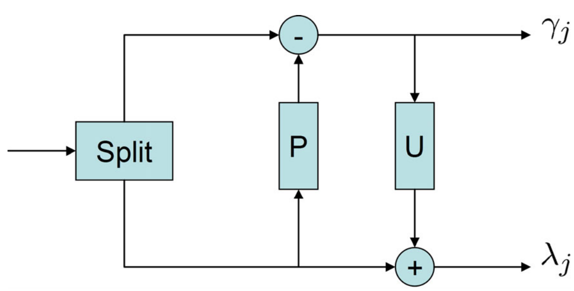

Figure 3 - Illustration of one step forward lifting

Forward and inverse S transformation equations for one lifting step are as below:

$$\text{Forward}: \begin{cases} y[2n+1] = x[2n+1] - x[2n] \\ y[2n] = x[2n] + \left\lfloor \dfrac{y[2n+1]}{2} \right\rfloor \end{cases}$$

$$\text{Inverse}: \begin{cases} x[2n] = y[2n] - \left\lfloor \dfrac{y[2n+1]}{2} \right\rfloor \\ x[2n+1] = y[2n+1] + x[2n] \end{cases}$$

where x, y[2n], and y[2n+1] are input, low-pass subband, and high-pass subband signal, respectively

Three lifting steps are employed for 8x8 blocks and two lifting steps are used of 4x4 blocks. To achieve that, illustration in Figure 3 is cascaded for approximation terms. Inverse lifting step is also obtained similarly.

2.2 CRFB Quantization Algorithm

A midtread quantizer is used for transform coefficients other than the lowest frequency coefficients of each CRFB block. The below equation defines the quantization process where *x* is the transform coefficient, *k* is and integer representing the quantization scale, sgn() is the signature function, and $\lfloor \cdot \rfloor$ represents rounding towards zero.

$$\mathrm{sgn}(x)\left\lfloor \frac{|x|}{2^k} + \frac{1}{2} \right\rfloor$$

Quantization can be implemented using just bit shifts since the scales are represented as powers of 2. In addition, a quantization matrix is used to allow finer quantization for lower frequency coefficients.

2.3 CRFB DC Prediction Algorithm

Proposed technique involves predicting lowest frequency transform coefficients (DC coefficient) of a CRFB block using its left, top and, top-left neighbor blocks. DC prediction is disabled for the IBDI-off mode to enable random access to 8x8 blocks. The prediction method is same as the MPEG-4 DC prediction [9].

Spatial prediction is performed separately for luminance and chrominance pixel blocks. Let *A, B and, C* be the DC coefficients of the top-left, top and, left neighbors of the current block to be encoded. The DC coefficient of the current block (*X*) is predicted as follows:

$$dir(X) = \min\left\{ |A-B|, |A-C|, \left| A - \frac{B+C}{2} \right| \right\},$$

$$pred(X) = \begin{cases} C, & \mathit{if}\ dir(X) = |A-B|, \mathit{or}\ B\ \mathit{is\ not\ available} \\ B, & \mathit{if}\ dir(X) = |A-C|, \mathit{or}\ C\ \mathit{is\ not\ available} \\ (B+C)/2, & \mathit{if}\ dir(X) = |A-(B+C)/2| \\ 0, & \mathit{if\ both}\ B\ \mathit{and}\ C\ \mathit{are\ not\ available} \end{cases}$$

The residual DC coefficient is computed by subtracting the prediction from the actual DC coefficient value. The decoder performs exact same prediction procedure and reconstructs the luma and chroma DC coefficients by adding the residual to the prediction.

2.4 CRFB Entropy Coding Algorithm

Exponential-Golomb (EG)[5] and Unary [6] variable length coding (VLC) techniques are employed. EG0 and EG3 codes are used to compress DC prediction residuals and quantized transform coefficients. Quantization indices are signaled after unary coding.

2.5 CRFB Encoder Design

Figure 4 shows the overall block diagram of the proposed CFB encoder.

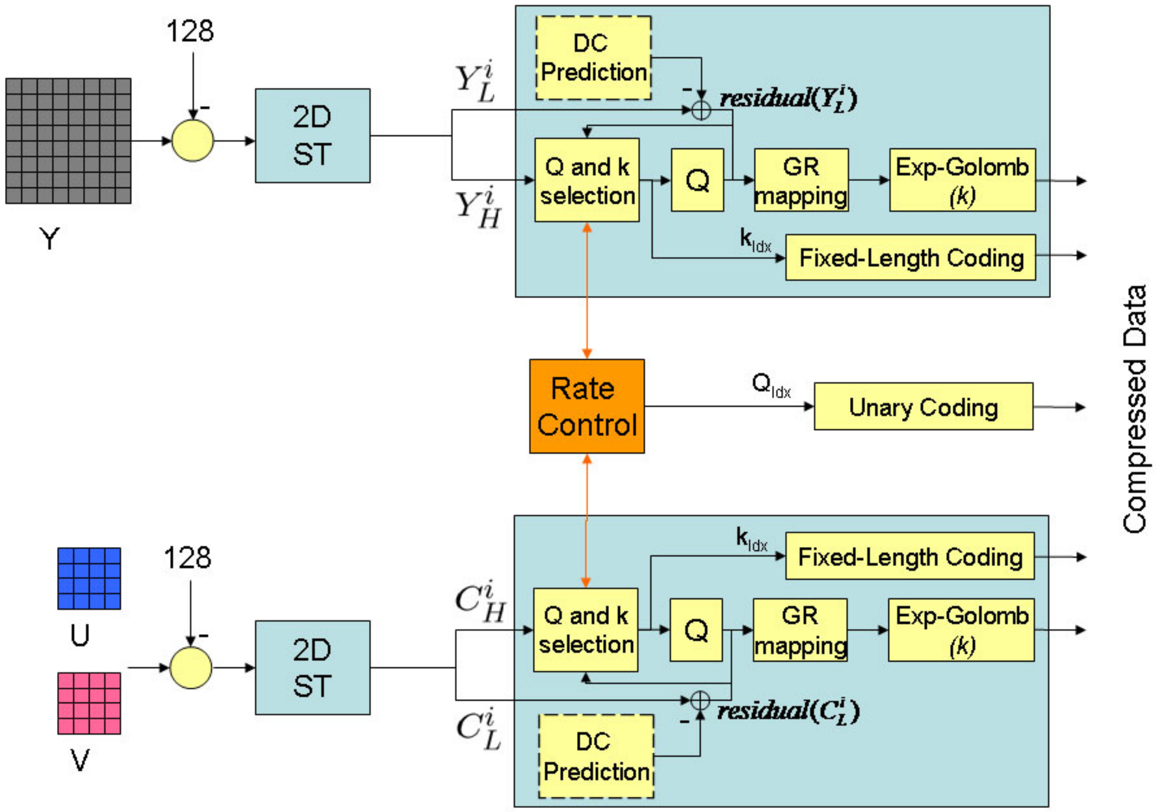

Figure 4 – Proposed CFB encoder block diagram

First, 2-D S transform of Y, U, and V pixel blocks are taken after shifting the pixel values. Second, transformed domain data is split into low and high frequency data, or equivalently called approximate and detail data. Spatial prediction is performed for low-frequency components using top-left, top and, left neighbor block DC coefficients and prediction residuals are computed (DC prediction is disabled for the IBDI-off mode). Prediction residuals go through GR mapping and EG coding without being quantized, due to their importance. GR mapping maps negative integers to positive odd integers and non-negative integers to positive even integers. For high-frequency components suitable Q is selected jointly with the k values that give the minimum total (low and high frequency coefficients) coded-length; after Golomb-Rice (GR) mapping, they are coded using EG(k).

2.6 CRFB Memory Access Unit (MAU) Size and Rate Control

Proposed algorithm guarantees fixed compression ratio for a memory access unit (MAU) to support random memory access in motion estimation and motion compensation. The size of the MAU is different for IBDI-on and IBDI-off modes.

2.6.1 IBDI-on Configuration

12 bit/pixel to 8 bit/pixel compression ratio is achieved for a block of pixels containing 8x8 Y, 4x4 U, and 4x4 V, when IBDI tool is turned on. The minimum quantization index satisfying this budget is selected by the rate control algorithm.

2.6.2 IBDI-off Configuration

The MAU is selected as a row of MBs (64 lines of luma, and 32 lines of chroma (U, V) pixels) for IBDI-off configuration to minimize the distortion caused by CRFB. A simple leaky-bucket like rate control technique is used to guarantee 8 bit/pixel to 4 bit/pixel compression. Unused bits after a CRFB block is compressed are transferred to the following CRFB blocks.

Accessing the memory in units of multiple pixel rows is particularly useful for a motion estimation architecture where strips of reference frame blocks are cached in on-chip memory. Such an approach is depicted in Figure 5. In this caching architecture, every pixel of the reference block is read once; hence the memory access bandwidth is inherently reduced, as discussed in [10]. Next MB row of reference pixels are loaded to the cache once the current row of MBs are processed. Limited vertical search range and larger local cache size requirements are the downsides of this architecture. However, same pixel location may need to be loaded multiple times in alternative architectures where smaller amount of reference frame data is cached.

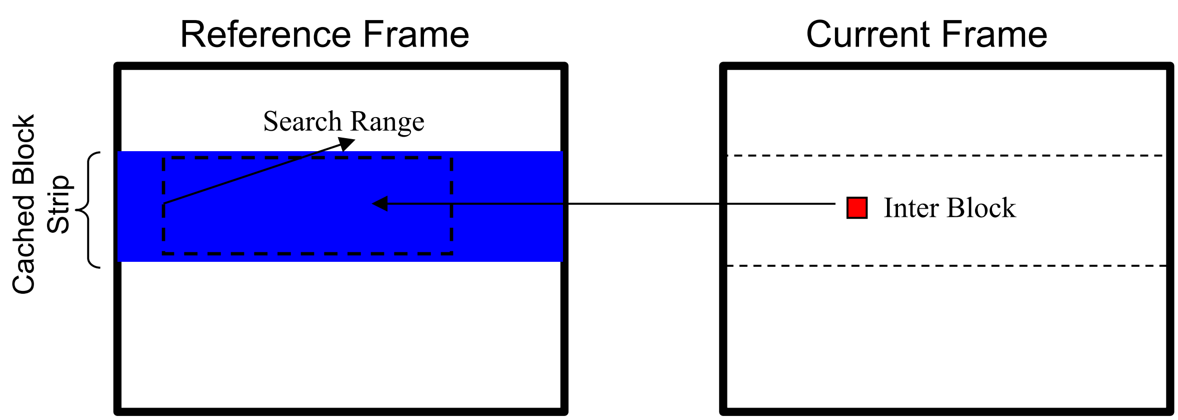

Figure 5 – Block strip based reference frame caching

3 Results

Reported results are verified as part of TE2. The summary of BD-Rate results compared to the unmodified KTA2.6r1 is as follows. Please refer to the attached spreadsheet for the detailed results.

3.1 IBDI-on

Table 1: CS1

	5-pt BD-Rate	Average BDLow+BDHigh	MAX	MIN
Class A	0.25	0.30	0.45	0.24
Class B	0.07	0.14	0.16	-0.02
Class C	0.07	0.12	0.16	0.01
Class D	0.23	0.17	0.63	0.00
All	0.13	0.16		

Table 2: CS2

	5-pt BD-Rate	Average BDLow+BDHigh	MAX	MIN
Class B	0.28	0.30	0.53	0.03
Class C	0.53	0.54	0.71	0.21
Class D	0.27	0.26	0.55	0.04
Class E	0.55	0.39	1.01	0.29
All	0.42	0.39		

3.2 IBDI-off

Table 3: CS1

	5-pt BD-Rate	Average BDLow+BDHigh	MAX	MIN
Class A	0.00	0.00	0.00	0.00
Class B	0.16	0.21	0.80	0.00
Class C	0.76	0.66	2.27	0.09
Class D	5.98	5.46	9.37	0.01
All	1.85	1.70		

Table 4: CS2

	5-pt BD-Rate	Average BDLow+BDHigh	MAX	MIN
Class B	0.09	0.11	0.45	0.00
Class C	1.81	1.72	6.16	0.08
Class D	5.66	5.34	10.25	-0.02
Class E	0.00	0.00	0.00	0.00
All	2.02	1.92		

4 Complexity

Average increase in encoding and decoding times compared to the KTA2.6r1 are reported. 64-bit Linux PCs with CPUs similar to Intel(R) Xeon(R) CPU X5570 (2.93 GHz) are used in the experiments.

4.1 IBDI-on

Table 5: CS1

	Decoder Run-time % Increase	Encoder Run-time % Increase
Class A	20.83	-2.28
Class B	15.19	-5.38
Class C	35.38	10.27
Class D	36.37	2.59
All	26.98	1.33

Table 6: CS2

	Decoder Run-time % Increase	Encoder Run-time % Increase
Class B	15.41	-6.32
Class C	31.73	-7.65
Class D	39.77	-5.50
Class E	23.82	-3.87
All	27.16	-5.99

4.2 IBDI-off

Table 7: CS1

	Decoder Run-time % Increase	Encoder Run-time % Increase
Class A	1.96	-10.50
Class B	6.70	-3.80
Class C	9.26	-1.85
Class D	11.55	-4.81
All	8.04	-4.45

Table 8: CS2

	Decoder Run-time % Increase	Encoder Run-time % Increase
Class B	4.82	-5.96
Class C	15.05	-2.47
Class D	11.30	2.22
Class E	13.80	-2.34
All	10.68	-2.36

5 Patent rights declaration(s)

Texas Instruments Inc. may have IPR relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

6 References

[1] M. Budagavi and M. Zhou, “Video coding using compressed reference frames”, ITU-T SG16/Q6, 31st VCEG Meeting, Marrakech, Morocco, January 2007, Doc. VCEG-AE19.

[2] M. Budagavi, V. Sze, M. U. Demircin, S. Dikbas, M. Zhou, A.P. Chandrakasan, “Video coding technology proposal by Texas Instruments”, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-A101, Dresden, Germany, April, 2010

[3] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” Appl. Comput. Harmon. Anal., vol. 3, pp. 186–200, Apr. 1996

[4] R. Calderbank, I. Daubechies, W. Sweldens, and B. Yeo, “Wavelet transforms that map integers to integers,” Appl. Comput. Harmon. Anal., vol. 5, pp. 332–369, July 1998

[5] J. Teuhola, "A Compression Method for Clustered Bit-Vectors", Information Processing Letters, Vol. 7, pp. 308-311, Oct. 1978.

[6] Sayood, K. “Introduction to Data Compression, Third Edition, (Morgan Kaufmann Series in Multimedia Information and Systems),” Morgan Kaufmann Publishers Inc, 2005.

[7] ITU-T SG 16/Q.6, “Draft Requirements for next-generation video coding project”, ITU-T SG 16/Q.6 Doc. VCEG-AL96, London, UK / Geneva, CH, 1-8 July, 2009.

[8] MPEG Video and Requirements Subgroups, “Vision, Applications and Requirements for High-Performance Video Coding (HVC),” ISO/IEC JTC1/SC29/WG11/N11096, Kyoto, JP, January 2010.

[9] ISO/IEC 14496-2, Information technology – coding of audio-visual objects – Part 2: Visual, 2001.

[10] Jen-Chieh Tuan; Tian-Sheuan Chang; Chein-Wei Jen; , "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," Circuits and Systems for Video Technology, IEEE Transactions on , vol.12, no.1, pp.61-72, Jan 2002