A Hybrid Hardware Architecture for LCD Overdrive Frame Buffer Reduction

Jiyong Choi, Ansun Jeong and Jungeun Baek
Display Solution Group, Dawin Technology Inc., Seoul, 135 - 504, Korea
TEL:82-2-529-2826, e-mail: jchoi@dawintech.com.

Keywords : Motion blur, LCD response time, Overdrive, Image Compression

Abstract

We present a hybrid hardware architecture capable of encoding and decoding a full HD resolution video with 60 fps. A number of technical modifications are applied to an existing image compression algorithm not only to accommodate large videos at a high frame rate but to enhance the quality of synthetic images, such as characters and video game images. Image quality of the proposed algorithm at a 1/6 compression ratio was comparable to that of the BTC based 1/3 compression algorithm.

1. Introduction

LCD panels are widely used in monitors, notebooks and TVs among other display solutions due to many of their attractive features. However, because of the slow response time and hold type display characteristic, LCD screens show motion blur when displaying fast moving images. A number of techniques such as backlight scanning [1] and overdriving [2] have been used to overcome this weakness. To reduce the overdrive frame buffer size, image compression algorithms like BTC (Block Truncation Coding) [3], scalable DCT (Discrete Cosine Transform) based algorithm [4] or color space conversion method [5] are used. High compression ratio algorithm is needed to support a full-HD video with less expensive frame memory.

The BTC based compression algorithm poses some fundamental limitations in increasing a compression ratio while DCT based compression algorithm has its own weakness in synthetic images such as the characters. The DCT based algorithm has its strength in the high compression ratio for natural images that are usually captured on digital video camcorders. However, in case of scrolling characters on various TV programs or fast-moving sharp edges of video games, DCT based algorithms for the overdrive frame buffer application are not well suited because of the blurring and ringing effects. Synthetic images are fundamentally different from natural images for their spatial high frequency component, hard predictability from the surrounding pixels especially for the characters, and repeatability of the color components in small areas. These underlying differences between the two image categories have led us to develop a hybrid architecture to improve the weaknesses seen in both digital images.

2. The Proposed Compression Scheme

Conventional DCT based compression scheme includes DCT, Scan, Quantize and Variable length coding blocks. A number of technical modifications were applied to this scheme to achieve a better image quality for both natural and synthetic images while maintaining moderate size hardware.

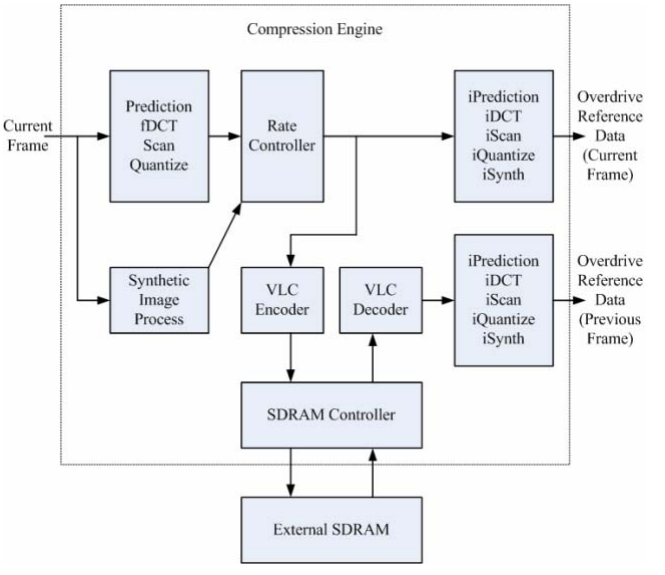

Fig. 1. Block diagram of Proposed Architecture

DCT based compression encoder necessarily requires reconstruction block which contains inverse prediction, inverse DCT, inverse Scan, and inverse Quantize block to generate decompressed data for the prediction block [6]. The reconstruction block can be regarded as a decoder. So this reconstruction block inside the encoder can act as both prediction data generator and extra decoder. This architectural characteristic makes the DCT based compression algorithm a natural candidate for the dual decoder e-FFD overdrive scheme [7] where the reconstruction block in encoder can function as one decoder. Figure 1 shows this architectural benefit.

The image quality is measured by calculating the PSNR of the decompressed image defined as follows:

$$MSE = \frac{1}{X \cdot Y} \sum_{x} \sum_{y} \left\| O(x,y) - R(x,y) \right\|^2$$
$$PSNR = 10 \cdot \log\left(\frac{255^2}{MSE} \right) \quad (1)$$

where X and Y are the number of horizontal and vertical pixels in the image, and O(x,y) is RGB pixel values at coordinate (x,y) in original input image, and R(x,y) is reconstructed pixel values after compression.

Macro-block size is carefully selected to avoid ringing effect, to minimize the number of image line buffers when implementing hardware, and to manage DCT calculation block size. The maximum number of non-zero coefficients after quantization is limited to the level at which compression image quality can be acceptable.

Dynamic color space conversion block determines whether the given image area's chroma information can be down-sampled or not. Only the chroma pixels in low spatial frequency area are down-sampled. This method helps to prevent blur and ringing of the strong edges which were big hurdles in YCbCr420 format DCT based compression algorithms.

DCT based compression algorithm requires variable length coding (VLC) scheme. This algorithm holds some difficulties when implementing with fixed size frame buffer because a rate control algorithm which is controlling quantize parameter value based on the feed back information from VLC encoder can not guarantee against buffer overflow. The raw image space is partitioned as segments which are mapped to the corresponding reduced memory space. Rate controller calculates the real code length before the VLC encoder actually generates codes for that block so that the buffer overflow can be surely avoided. General hardware implement use double segment buffer to accumulate compressed variable length codes before writing to external SDRAM in burst. Triple segment buffer is used in this architecture to make sure of smoothness of image quality when crossing the segmented memory boundaries.

Multiple variable length encoders and decoders operate in parallel to process broad bandwidth and occasionally burst variable length data at a low clock rate. The compressed data structure is packetized and organized as linked lists so that multiple channel data could be concurrently processed by each of VLC encoders and decoders. As a result, single external memory can be viewed as multiple memories from the each VLC codec. A new set of VLC table has been created such that maximum symbol length is limited to downsize hardware and code length can be simply calculated by rate controller. Also the code table is optimized for the 1/6 or lower compression ratio.

In order to enhance synthetic image quality after compression, a dedicated synthetic image processor detects sharp edge from the input image by analyzing the low and high groups of each pixel's luminance levels and creates 'silhouette' bit map for that macro-block. The decision regarding whether the macro-block is natural or synthetic is made based on the gap between the two levels and noise of the each pixel group. First order differential algorithm is used for better noise immunity. The bit map together with low and high levels is coded for the synthetic image. The low and high levels of color components in the synthetic macro-block are tend to be correlated with surrounding blocks. This relationship can be exploited by storing these two levels into the local color palette for the adjacent or near blocks so that further compression can be achieved.

3. Results and discussion

The compression algorithm that we suggest has two advantages. One is that the input video clock can be used as internal operating clock. So no extra clock sources like PLL is necessary. The other is that it can compress images up to 1/6 of their original size while preserving the image quality to that of the BTC based algorithm at 1/3. Embedded frame memory could be a feasible solution at this compression ratio. This becomes more noticeable advantage in the higher resolution displays from the cost effectiveness point of view.

We used Lena and Baboon images for the test, which are generally used in image processing fields. Also, to evaluate the performance of the algorithm in synthetic images, we used 'Master Image'. Table 1 compares image qualities among different compression schemes.

TABLE 1. Image Quality (PSNR)

	YCbCr422	BTC based Compression	Proposed Algorithm
Lena	37.6 dB	34.4 dB	33.9 dB
Baboon	41.9 dB	30.7 dB	33.1 dB
Master	37.1 dB	28.9 dB	40.4dB

Fig. 2 to Fig. 4 are quoted from the 'Master image' to illustrate synthetic image quality after compression. The Fig. 3 is YCbCr420 color space conversion which could be considered as a 1/2 compression method. The Fig. 4 is BTC based 1/3 compression algorithm and Fig. 5 is the proposed 1/6 compression algorithm.

Fig. 2. Original

Fig. 3. YCbCr420

Fig. 4. BTC Based

Fig. 5. Proposed

The BTC based compression algorithm shows blocky effect and color leakage around the edges while YCbCr420 shows yellow color fade. The proposed hardware maintains the left edge of the character by 'silhouette' mode compression while keeping smoothness in shadow area by DCT mode compression. The Fig. 6 is Lena original image and Fig. 7 is reconstructed image. Fairly good image quality of Fig. 7 highlights the strong point of DCT based compression in natural images.

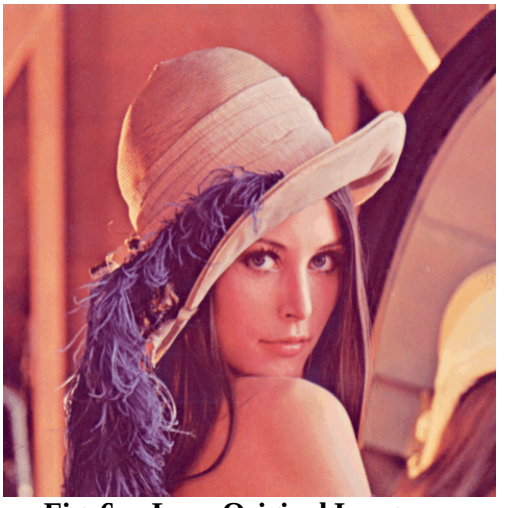

Fig. 6. Lena Original Image

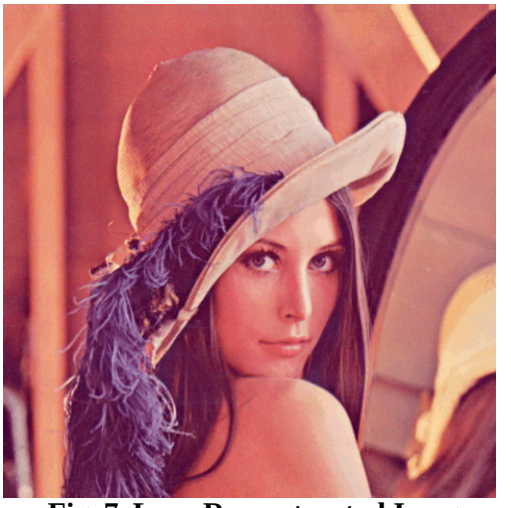

Fig. 7. Lena Reconstructed Image (After compressed by proposed scheme)

The compression engine is implemented on Altera Stratix® FPGA with external SDRAM as frame buffer. Video clock was used as internal operating clock as well as SDRAM interface clock so that the entire system could be working in single clock domain.

4. Conclusion

This technique can be applied to various real-time applications such as LCD over-drive frame buffer reduction. Specifically, it is beneficial in reducing the number of bits in embedded frame buffer. Hence a die size reduction would be achieved. The proposed scheme allows variable compression ratio within the image quality tolerance for other applications. Study is under way to further expand this architecture to handle full-HD video at 120Hz for the next generation LCD overdrive application.

5. References

1. Jun-ichi Hirakata et al., SID'01 Digest, pp 990-993 (2001)
2. Baek-woon Lee et al., SID'01 Digest, pp 1260-1263, (2001)
3. J. Someya et al., SID'03 Digest, pp 1346-1349 (2003)
4. Vlenten, R. J. van der et al., Int. Conf. Image Proc. (ICIP 2000), vol. III, (Vancouver, Canada), pp. 837-840, Sep. 10-13, 20
5. Sung-Jo Koo, Chang-Gon Kim et al., SID'05 Digest, pp 474 - 477 (2005)
6. Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression"
7. J. Someya et al., SID'02 Digest, pp 72 - 75 (2002)