A hybrid image coding in overdriving for motion blur reduction in LCD

Jun Wang， Kyeongyuk Min, Jongwha Chong

Hanyang University, Seoul, Korea, 133-791
junmei0073@hotmail.com, kymin@hanyang.ac.kr, jchong@hanyang.ac.kr

Abstract. Overdriving technique enlarges the desired change of the pixel value，the error in general compression methods is enlarged at the same time. Therefore， we propose a novel Adaptive Quantization Coding (AQC) to reduce the error in compression for overdriving technique reducing motion blur. Considering hardware implementation, we develop a hybrid image coding which uses color transform first, and then uses AQC to compress luminance data as well as Block Truncation Coding (BTC) to compress chrominance data. The simulation results shown that the average PSNR was improved 5.676dB as compared with the result of BTC, and the average SD of error was reduced 50.2% than that in the BTC. The proposed algorithm is implemented with the verilog HDL and synthesized with the synopsys design compiler using 0.13μm Samsung Library.

Keywords: LCD, Motion blur, Overdriving, Compression, Block Truncation Coding, Adaptive Quantization Coding.

1 Introduction

Nowadays, Liquid Crystal Displays (LCDs) have been widely considered to have many advantages over Cathode Ray Tube (CRT) displays in the respects of resolution, power consumption, size, thickness, and other critical parameters. However, LCDs have a drawback of motion blur in TV applications and desktop for entertainment. One of the causes of motion blur is the sample-and-hold characteristic of the LCD panel, i.e. the image during the frame time is continuously shown. This type of motion blur can be reduced with a flashing or scanning backlight as in [1]. Another cause of motion blur is the slow reaction of the liquid crystal (LC) cell to a change in the pixel value as in [2]. Strong user demand for a high performance display creates the need to rapidly improve the liquid crystal response time and minimize the motion blur. Therefore, a lot of efforts have been put into speeding up the response of LC materials. This can be done by applying better materials, or by improving LC cell design as in [3]. There is also a well known method for response time improvement based on video processing called overdrive as in [4]. The technique of overdrive reduces motion blur by enlarging the desired change in the pixel value in order to force the LC material to react faster as in [5].The illustration of overdrive mechanism in LCD is shown in figure 1.

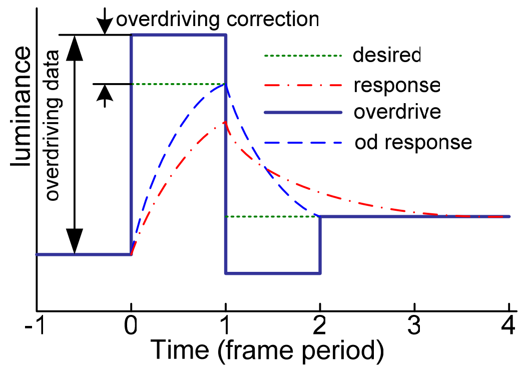

Fig. 1. Illustration of overdrive mechanism in LCD.

Since overdrive compensates for a non-linear effect of the LCD panel it should preferably be placed at the end of the video processing chain, right before the LCD panel as in [2]. According to the pixel values of the current frame and the previous frame, the look-up table (LUT) provides the overdriving data to the LCD panel. The frame memory stores the pixel values of the current frame and give an output of the pixel value of the to the lookup table. A simple block diagram of general overdriving circuit is shown in figure 2.

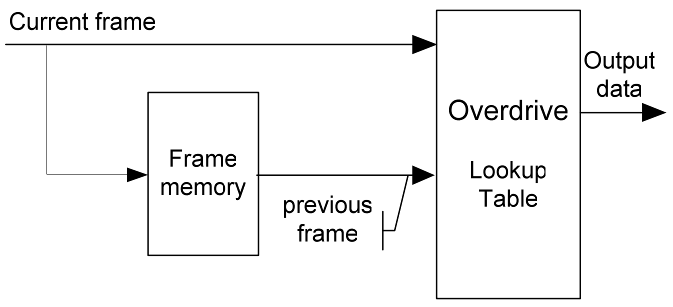

Fig. 2. Block diagram of overdrive.

However, the conventional overdriving technique stores the current frame in the dynamic random access memory (DRAM) which is large and expensive. It is desired to reduce the cost of the system by compressing the data stored in the DRAM as in [6] and [7]. The errors in the decompressed image cause the errors in the amount of overdriving correction and have an effect on the reduction of motion blur. It is necessary to develop an effective compression method to reduce the errors in the decompressed image by the general compressing methods such as quantization, sub-sampling and block truncation coding (BTC). Therefore, we propose a novel hybrid image coding (HIC) to reduce the errors in decompressed image for the reduction of frame memory in LCD overdriving. The proposal uses a color space transform first, and then the proposed adaptive quantization coding (AQC) to compress the luminance data as well as the BTC to compress the chrominance data.

This paper will describe the background of the BTC techniques in Section 2.1. A novel proposal for the adaptive quantization coding (AQC) techniques is introduced in Section 2.2. In Section 2.3, a hybrid image coding method which integrates the advantages of the AQC and the BTC is introduced. In Section 3, the simulation result is described. This paper will be concluded in Section 4.

2. Proposal

2.1 The background of the BTC

Block Truncation Coding (BTC) is a type of lossy compression technique for grayscale images. It divides the original images into small sub-images and then uses the quantization method, which adapts itself to the image statistics to reduce the number of gray levels in the image.

The compression procedure is as follows:

An image is divided into blocks of 4 x 4, 4 x 2, 4 x 1 or 2 x 2 pixels. For each block, the mean is computed; the value changes from block to block. Then a two-level quantization in the block is made as follows: If the value of a pixel is greater than the mean it is offered at a value of "1"; otherwise, it will be "0". By this way, the block is divided into two groups: upper group and lower group. The mean of the upper group and the mean of the lower group are computed. The image is reconstructed with the upper group mean, the lower group mean and the bit-plane which is made up of “1” or “0”.

2.2 The proposed Adaptive Quantization Coding (AQC)

The proposed coding method Adaptive Quantization Coding (AQC) is also a type of lossy image compression technique for grayscale images. It divides the original images into small sub-images and then uses an adaptive quantization, which adapts itself to the image statistics to reduce the number of gray levels in the image.

The compression procedure is as follows:

An image is divided into blocks of 4 x 4, 4 x 2 or 2 x 2 pixels. For each block, the minimum and the maximum are computed; these values change from block to block. Then an eight-level quantization in the block is made and the quantization step adapts itself to the difference of the block. So the value of the pixel is offered from "000" to "111" according to the level which the pixel belongs to. The image is reconstructed with the minimum, the quantization step and the bit-plane which is made up of three binary bits.

2.3 The hybrid image coding method

The block diagram of overdriving circuit using compression is shown in figure 3. The current frame in 24bit/pixel is compressed by the encoder and conserved in frame memory. The decoder gets data from frame memory and gives an output of decompressed previous frame. The module of overdriving lookup table uses the lookup table and gives an output of overdriving data according to the pixel value of the current frame and the decompressed previous frame.

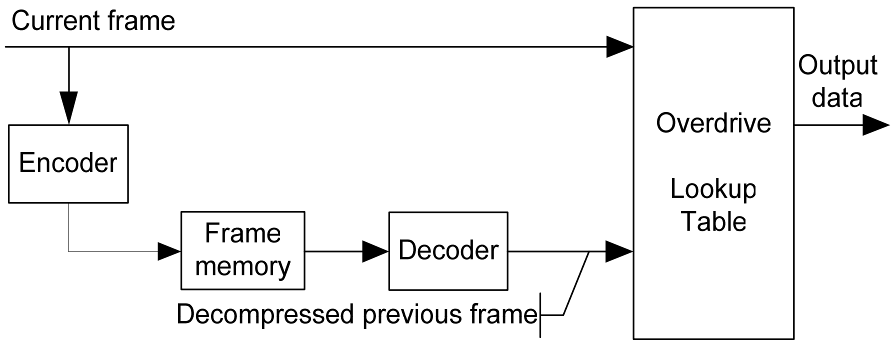

Fig. 3. Block diagram of overdriving circuit using image compression.

The human visual system (HVS) is less sensitive to color than to luminance (brightness), so that we have proposed an algorithm of HIC. The HIC converts RGB color space into YCbCr color space, first. Then it uses an efficient coding method of the proposed AQC to compress the luminance data which is more sensitive to HVS than color as well as the conventional simple method of BTC to compress the chrominance data. The YCbCr sampling format is 4:2:0 which is exactly half times as many samples as R: G: B video. The block diagram of the encoder and the decoder in the HIC is shown in figure 4.

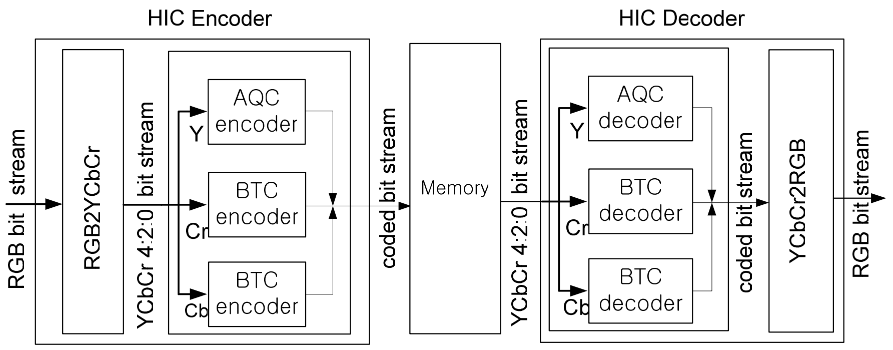

Fig. 4. Block diagram of encoder and decoder in the HIC.

In the *HIC encoder* shown in figure 4, there are 4 modules: *RGB2YCbCr* module, the *AQC encoder* module and two *BTC encoder* modules. In *RGB2YCbCr* module, it converts the frame data from RGB color space into YCbCr color space, and gives us an output of YCbCr 4:2:0 bit stream in the YCbCr sampling format. In the *AQC encoder* module, it uses the AQC to encode the luminance component *Y*. In the two *BTC encoder* modules, they use the BTC to encode the chrominance component *Cr* and *Cb*, respectively. The output of the AQC and the BTC modules are stored in the reduced frame memory.

In the *HIC decoder* shown in figure 4, there are 4 modules: *AQC decoder* module, two *BTC decoder* modules and *YCbCr2RGB* module. In the *AQC decoder* module, it uses the AQC to decode the luminance component *Y*. In the two *BTC decoder* modules, they use the BTC to decode the chrominance component *Cr* and *Cb*, respectively. In *YCbCr2RGB* module, it converts the frame data from YCbCr color space into RGB color space, and gives us an output of RGB bit stream which will be supplied to the overdriving lookup table module.

The block size can affect not only the coding performance but also the compression ratio. So, to determine the block size, we must investigate the relationship among the coding performance, the compression ratio and the block size. The coding performance and the compression ratio of the AQC and the BTC in different block sizes in case of luminance data Y are shown in table Ⅰ. The PSNR and the SD of error are used to evaluate the image coding performance. The test image is “Lena” which is used generally.

Table Ⅰ. Compare the AQC coding performance and the compression ratio with the BTC in different block sizes in case of luminance component Y.

Lena	PSNR (dB)		SD		Compression ratio	
Block size	BTC	AQC	BTC	AQC	BTC	AQC
8 x 8	31.385	41.060	6.858	2.242	6.40	2.49
8 x 4	32.289	42.320	6.177	1.935	5.30	2.33
4 x 4	34.608	45.100	4.724	1.392	4.00	2.06
4 x 2	36.089	47.133	3.983	1.100	2.67	1.68
2 x 2	41.266	52.053	2.191	0.622	1.60	1.23
4 x 1	37.624	49.153	3.343	0.875	1.60	1.23

From table Ⅰ, three trends can be seen in the following:

1. The coding performance of the AQC and the BTC become better and better when block size becomes smaller and smaller.

2. The compression ratio of the AQC and the BTC become smaller and smaller when block size becomes smaller and smaller.

3. The AQC gains a much better coding performance in case of every block size.

Before coding by the AQC and the BTC, the data is transformed from RGB format to YCbCr 4:2:0 format, and the compression ratio in the course of color transformation is 2. The preferred option is to compress the data of YCbCr 4:2:0 with compression ratio slightly more than 1.5 in order to get better image quality.

Furthermore, to reduce the system cost by using one line buffer memory in hardware implementation, the blocks of 4 x 2 AQC, 2 x 2 BTC and 4 x 1 BTC are the preferred candidates, and the performances of these are descended in turn in case of similar compression ratio.

To apply the 4 x 2 block AQC which is the best of the three candidates to the more sensitive luminance data, the chrominance data Cr and Cb must be compressed in 4 x 1 block BTC considering the hardware implementation. As the image data is converted from RGB format into YCbCr 4:4:4 format, the YCbCr 4:4:4 format data are sub-sampled to YCbCr 4:2:0 format data. In YCbCr 4:2:0 sampling format, the chrominance data Cr and Cb have only half as many samples as Y. For chrominance data, therefore, the 4 x 2 block AQC or the 2 x 2 block BTC can not be used and the only available candidate is to use 4 x 1 block BTC. Finally, the HIC can gain a compression ratio of 3.31.

The proposed algorithm is implemented with software in C language and with hardware in the verilog HDL and synthesized with the synopsys design compiler using 0.13μ m Samsung Library. The delay of encoder is 34 clocks and the delay of decoder 11 clocks.

3. Results

If image compression is used to reduce the image data stored in the frame memory, the reconstructed image will contain errors. In the configuration shown in Figure 3, the encoded image data is used only to detect the amount of temporal change between the current frame and previous frame and the encoded image data is never displayed directly. Errors caused by reducing the image data become errors of the overdriving correction, and lead to the insufficient or excessive correction of response time. Therefore the performance of overdriving with memory reduction can be evaluated to investigate the amount of errors in the decompressed image. We can evaluate the total error by the PSNR which is well known as an evaluation method of image coding, but we also care for the distribution of the error. That can be evaluated by the SD of the error.

To evaluate the proposed algorithm of the AQC and the HIC, the following general test images are used for simulation: ‘Airplane’, ‘Baboon’, ‘Lena’, ‘Sailboat’, ‘Tiffany’. The simulation has been carried out in three coding methods: the BTC uses the BTC of 4 x 2 block with compression ratio of 2.67 to compress the RGB data; the TBC uses 2 x 2 BTC to replace 4 x 2 AQC basing on the HIC; the HIC is the proposed algorithm. The compression ratios are 2.67, 3.2 and 3.31, respectively. The figure 5 shows the simulation result of PSNR, and the figure 6 shows the simulation results of SD of error.

In figure 5, we can see that the PSNR of the HIC is higher than that of the TBC and the PSNR of the TBC is higher than that of the BTC in case of every test image, though the PSNRs of each method are different in these images. The average PSNRs of these samples are 30.864dB, 33.611dB and 36.540dB for the BTC, the TBC and the HIC, respectively. It is said that the TBC which uses the BTC to compress the data of converted YCbCr 4:2:0 color space can gain an improvement of 2.746dB in PSNR as compared with the BTC which compresses the data of RGB color space. The performance can improve 2.929dB in PSNR than the TBC using 4 x 2 AQC to replace 2 x 2 BTC in case of the HIC. Totally, the PSNR of image compressed in the HIC is improved as much as 5.676dB than that of the BTC.

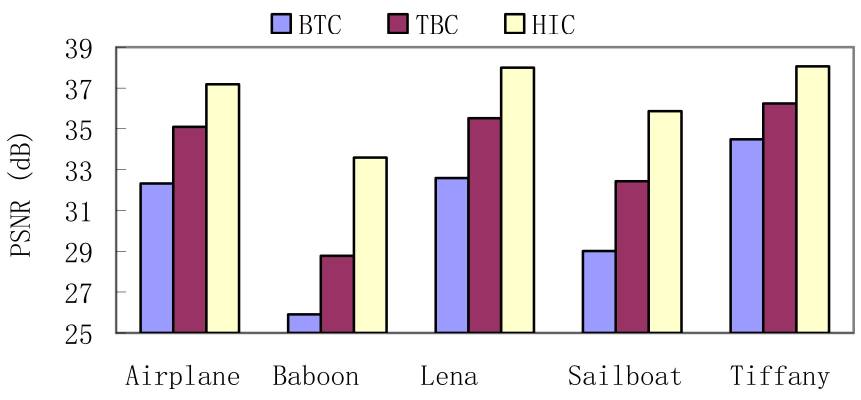

Fig. 5. The PSNR of various test images.

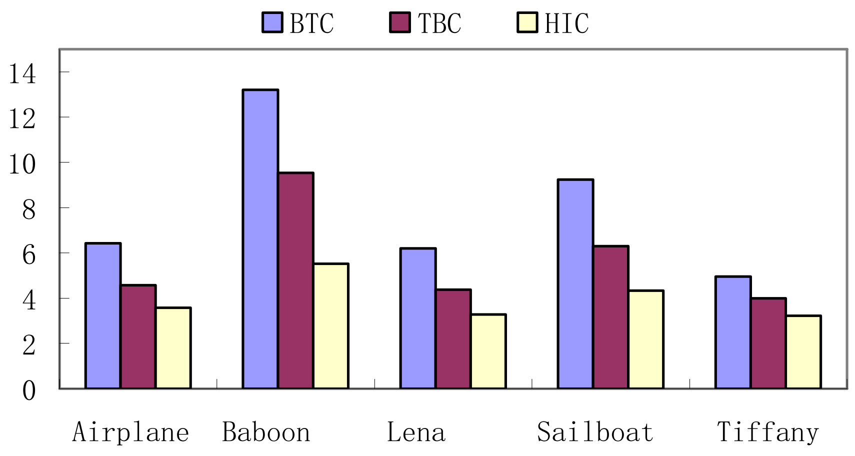

Fig. 6. The SD of various test images data error.

In figure 6, we can see that the SD of error in decoded image used the HIC is smaller than that of the TBC and the SD of the TBC is smaller than that of the BTC in case of every test image, though the SD of each method are different in these images. The average SDs of these samples are 8.006, 5.756 and 3.987 for the BTC, the TBC and the HIC, respectively. It is said that the TBC which uses the BTC to compress the data of converted YCbCr 4:2:0 color space can reduce 28.1% in SD comparing with the BTC which compress the data of RGB color space. Because of using 4 x 2 AQC to replace 2 x 2 BTC in the HIC, the performance can reduce 30.7% in SD than the TBC. Totally, the SD in the HIC can reduce 50.2% than that in the BTC.

4. Conclusion

In this paper, we proposed a novel Adaptive Quantization Coding to reduce the error in compression for overdriving technique reducing motion blur. Considering hardware implementation, we developed a hybrid image coding which uses color convert first, and then uses the AQC method to compress luminance data and uses the BTC to compress chrominance data. The simulation results shown that the average PSNR was improved 5.676dB as compared with the result of the BTC, and the average SD of error was reduced 50.2% than that in the BTC. The proposed algorithm was implemented with the verilog HDL and synthesized with the synopsys design compiler using 0.13μ m Samsung Library. The proposed algorithm efficiently reduced 30.2% of the image data stored in the frame memory. It reduced the error in decompressed image significantly and got more accurate overdriving data. The technique can be used to reduce the motion blur in TV applications and desktop for entertainment. In the future we will study the relationship between image features and errors using image samples such as graphic images and evaluate some other image compression methods.

Acknowledgments. We wish to thank the following colleagues for their cooperation in hardware design: Seungman Pyen for the AQC encoder module, decoder module, RGB2YUV module and YUV2RGB module; Dongsu seo for the BTC encoder module and decoder module.

References

1. Shimodaira, Y.: Invited Paper: Fundamental Phenomena Underlying Artifacts Induced by Image Motion and the Solutions for Decreasing the Artifacts on FPDs. SID. 03 Digest (2003) 1034-1037
2. R. H. M. Wubben, G. J. Hekstra: LCD Overdrive Frame Memory Reduction using Scalable DCT-based Compression. SID. 04 DIGEST (2004) 1348-1351
3. Michiel A. Klompenhouwer and Leo Jan Velthoven: LCD Motion Blur Reduction with Motion Compensated Inverse Filtering. SID. 04 DIGEST (2004) 1340-1343
4. H. Okumura: A new low-image-lag drive method for large size LCTVs. Journal of the SID. 1(3) (1993) 335-339
5. Hartman, R.A. et.al.: Fast Response Electro-Optic Display Device. United States Patent US5, 495-265
6. Jun Someya et.al.: Reduction of Memory Capacity in Feedforward Driving by Image Compression. SID. 02 Digest (2002) 72-75
7. Jun Someya et al.: A new LCD Controller for Improvement of Response Time by Compression FFD. SID. 03 Digest (2003) 1346-1349