Effective Compression Method for De-Burn-In Data

Shumeng Ding *, Wenhui Yu*, Jianhua Liang*, Qiqiang Han*, Yingying Tang* and Yaocheng Liu*

*** Goertek Co., Ltd. Xi’an city, China**

Abstract

De Burn-in (DBI) based on data accumulation can effectively alleviate screen aging phenomenon. However, it will produce large amounts of data. Since DBI has tolerance to data errors, a specialized compression method is proposed. This method can flexibly select the compression ratio according to different scenarios, greatly saving hardware resources.

Author Keywords

De Burn-in data compression; lossy compression; discrete cosine transform (DCT); color-space conversation (CSC); Rate Control (RC).

1. Introduction

An Organic Light-Emitting Diode (OLED) screen has the advantages of high brightness, wide color gamut and pure black display, but it also faces a more serious aging problem [1]. It appears as a visual artifact and this problem is called as ‘burn-in’. In order to alleviate the ‘burn-in’, the DBI techniques are applied.

DBI techniques that based on avoidance can extend life time of OLED, but often leads to a poor viewing experience [2]. To obtain a better viewing experience, techniques that based on compensation are proposed [2~4]. Among them, DBI based on data accumulation [5] has been widely utilized because it can keep panel design low cost. During its compensation process, a mass of data is produced to indicate how much the OLED have aged. If the data can be compressed, the hardware resources can be greatly saved.

However, for DBI based on data accumulation, the accumulation process will cause error propagation during compression, which will introduce large uncertainties. At present, there are few effective compression methods for this data.

To get pass this bottleneck, the DBI data is analyzed in detail, and it has the following characteristics:

- Only after a long period of accumulative use can the aging degree of the OLED change. Therefore, DBI has tolerance to data errors, and in different accumulation stages, DBI have different tolerance to errors.
- The DBI data increases monotonically with time, and the data range will change as accumulation stage changes.
- The non-uniformity of DBI data tends to increase over time. A larger DBI data indicates that the screen at the corresponding position is aging more seriously.
- The DBI data is spatially correlated.
- The rate at which the DBI data corresponding to each point increases and the final data value distribution are closely related to the content played by the user.

The first characteristic indicates that the DBI data has compression potential. It can be drawn by the second characteristic that the error of compression will propagate temporally. Therefore, it is necessary to prevent errors from always accumulating in one direction, thereby reducing the accumulative error. According to the third characteristic, if the same number of bits is allocated to blocks of different numerical sizes during the coding process, the data at severely aged locations will face more serious errors. To ensure the compensation performance of DBI, the rate control (RC) mechanism [6] is introduced to allocate different bits to different blocks. From the fourth characteristic, Method used to reduce space redundancy should be applied. The last characteristic presents that the distribution characteristics of data are uncertain, so multiple bit rates are set to cope with different scenarios.

Based on the data characteristics, this paper proposes a specialized compression method for DBI data. The adjustable compression bit rate ensure that the method can hold different scenarios, and save the storage effectively.

2. Principle

Figure 1 illustrates how the compression method is linked with DBI.

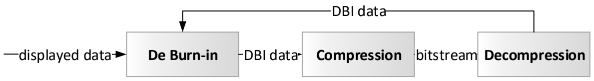

Figure 1. The linkage between DBI and compression.

Figure 2 shows the flow of the compression and decompression algorithm, and the algorithm s done at the slice level. Since the proposed algorithm is consistent with DBI data characteristics, a smaller slice can be adopted. When compression and decompression require simultaneous GRAM access, a smaller slice can consume less memory.

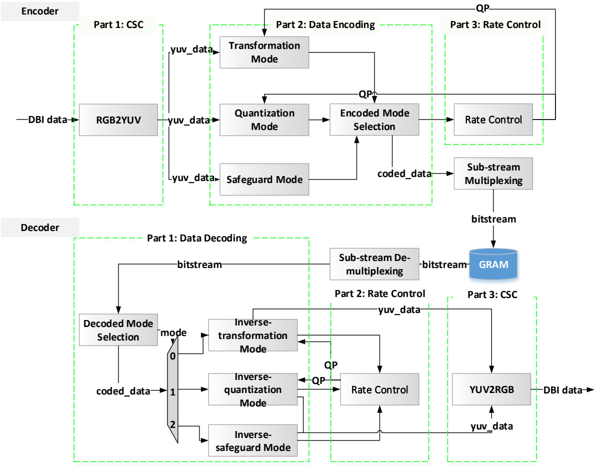

Figure 2. The flow of the compression and decompression.

As shown in Figure 2, there are three parts in compression and decompression respectively. Decompression is the inverse process of compression. Therefore, the following describes only the compression process in detail.

Part 1: Color-space Conversation

ISSN 0097-996X/25/5603-1693-$1.00 © 2025

The CSC is applied to convert DBI data in the RGB color space to YCoCg domain. This step can effectively remove the correlation between RGB channels, thereby improving the compression performance. The conversion formula is as follows:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 2 & -1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

Part 2: Data Encoding.

In this part, three encoding modes are used to encode the data, and the most appropriate encoding method is selected for the current block. The processing unit of this part is the 1x8 block. The spatial correlation between the data weakens over time, so the size of this block is suitable for DBI data compression.

For transformation mode, the first step is to transform data from the spatial domain to the frequency domain by DCT. This step can effectively remove the spatial correlation between data, and the transformation formula applied is as follows:

$$coeff(\mu) = A(\mu) \sum_{n=0}^{N-1} T(n) \cos(\frac{(2n+1)\mu\pi}{2N}) \tag{2}$$

$$A(\mu) = \begin{cases} \sqrt{\frac{1}{N}}, & \mu = 0 \\ \sqrt{\frac{2}{N}}, & \mu \neq 0 \end{cases} \tag{3}$$

Where $\mu = 0, 1, 2 \ldots, N-1$, $N = 8$ and n is the horizontal position of each pixel.

Then, the result (coeff) of the transformation is quantified. In this step, the quantization level depends on the quantization parameter (QP) calculated by RC module. The larger the QP, the greater the quantization level, and the smaller the number of encoded bits required by the current block. The quantization formula is as follows：

$$a = \frac{coeff}{|coeff|} \tag{4}$$

$$q\_coeff = a \times (coeff + dead\_zone) \gg QP \tag{5}$$

Increasing the value of a will cause more positive errors. On the contrary, there are more negative errors. Therefore, the quantization formula avoids unidirectional error accumulation by adjusting the value of dead_zone adaptively.

After the transformation, the number of data with smaller values is often greater than the number of data with larger values. Thus, the last step of transformation mode is to encode q_coeff by an adaptive Exponential-Golomb coding. By assigning a shorter code length to small data and a longer code length to large data, this step can further reduce coding redundancy. Considering the range of DBI data varies significantly over time, the parameter k of Exponential-Golomb coding also changes to improve compression performance. The larger the data range, the larger the k value.

Quantization mode is used to ensure that the length of the encoded data does not exceed the length of the original data. For this mode, the data after CSC is quantified directly by QP calculated by RC, and the output is the quantified data.

If too many bits are allocated to encoded blocks, subsequent blocks will not be encoded properly. In order to ensure that the entire process can always operate normally, safeguard mode is applied. This mode quantifies the data after CSC with a fixed value, so that the average encoding length of the output result is always lower than the pre-set value.

After all modes have encoded the data, select which coding mode is used to encode the current block. Generally, the transformation mode is used. If the encoding length of transformation mode exceeds the length of the original data, quantization mode is selected. If the transformation mode results in not enough bits left for subsequent blocks, safeguard mode is applied.

Part 3: Rate Control.

The RC module is applied to ensure that the compressed data of each slice is output at a fixed bit rate by adjusting QP for each block. The QP is calculated implicitly and identically by both the encoder and decoder, based on the RC state, so the QP does not need to be transmitted in the bitstream.

The first step is to build a buffer model to temporarily store the encoded results. Remove bits from the buffer at a constant rate (average_bits) while storing the encoded result (coded_bits) of the current block into the buffer. Thus, the fullness of this buffer is calculated as:

$$buffer\_fullness = buffer\_fullness + coded\_bits - average\_bits \tag{6}$$

As can be seen from the above formula, if the buffer fullness increases, it means that the encoding amount of the current block is greater than the average_bits. If the amount stored is reduced, the reverse is true. Therefore, fullness of the buffer can reflect the bit allocation of the encoded block. Otherwise, the encoder must ensure that this rate buffer never underflows or overflows.

Next, the ideal number of bits (target_bits) that should be allocated to the current block is calculated. The calculation is based on the number of unused bits (remain_bits) and the number of unencoded blocks (remain_blks).

$$target\_bits = \frac{remain\_bits}{remain\_blks} \tag{7}$$

Assuming that the adjacent blocks are similar, the QP (pre_qp) and the amount of encoding bits (pre_bits) of the previous block can be used as the base to calculate the QP of the current block.

Finally, pre_qp is adjusted according to buffer_fullness and target_bits to obtain the QP of the current block. If the buffer fullness is too large, QP is increased. Conversely, reduce QP. If the pre_bits is much larger than the target_bits, QP is increased. Conversely, reduce QP.

In addition, Sub-stream Multiplexing (SSM) is adopted to increase the throughput of the method.

3. Simulation Experiments and Results

Simulation experiment were conducted to verify the performance of the proposed compression method. The produces involved in this simulation experiment are summarized in Figure 3. The simulation experiment consists of two parts: the first part is the simulation experiment performed on linkage of DBI, compression and decompression module, and the second part is the simulation experiment performed on DBI only.

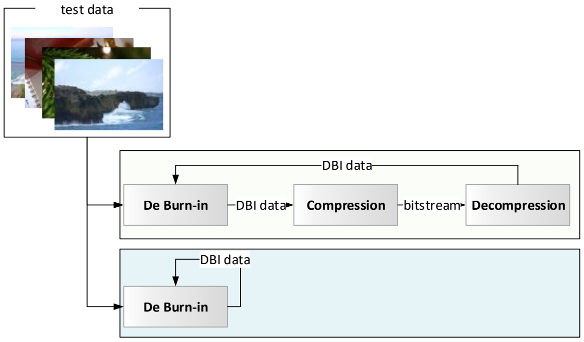

Figure 3. The produces involved in simulation experiment.

For the first part, the DBI module reads the data displayed on the screen and produces the data to be compressed. Then, the compression module is applied to compress the data. Finally, the bitstream is decompressed to be used by DBI in next turn. For the second part, the DBI module reads the same data as the first part, and produces the data to be used in next turn.

Considering that playing sticking content and background fixed content are more likely to produce aging phenomenon on the screen, the test data is mainly divided into three types of data as shown in the Figure 4: sticking video, common video, and live video.

Figure 4. The test data used in simulation experiment.

As shown in Figure 5, the DBI data will reach a larger data range with the accumulative time increases, so the compression performance under different accumulation time is tested. The accumulative duration of the simulation experiment is 100h, 400h and 1000h, and the compression ratios tested are 2.0x and 2.6x. By comparing the output compensation results of the two parts in Figure 2, the performance of the compression algorithm can be evaluated. Note that in order to see the compensation effect more clearly, the simulation experiment is based on the assumption that the OLED has undergone a certain degree of aging.

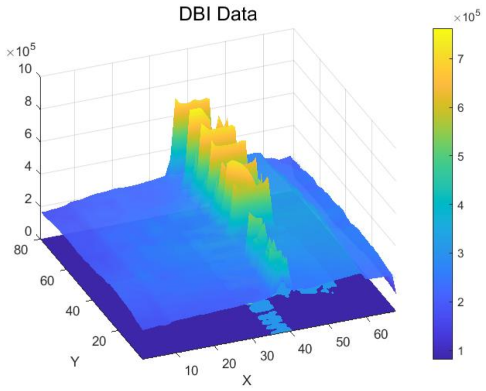

Figure 5. The DBI data after accumulation.

The evaluation indicators include objective indicators and subjective indicators. Because DBI data is monotonically increasing, the data range will increase with time, so Peak Signal to Noise Ratio (PSNR) [7] is inapplicable to evaluate the compression effect as an objective index. The data error of DBI does not exceed a certain percentage, it will not cause the difference in compensation results. Therefore, the percentage of error (p_error) is used as an objective indicator:

$$p\_error = \left| \frac{data\_comp - data\_ground}{data\_ground} \right| \times 100\% \qquad (8)$$

Where data_comp is DBI data with compression, and data_ground presents DBI data without compression.

The subjective index is to directly compare whether the compensation results of the two parts are visually different.

Table 1 shows the p_error of three data sets at different ratios and accumulative duration.

Table 1. The percentage of error of proposed method.

Data set	Ratio	Output bpc	Percentage of error		
			100h	400h	1000h
sticking video	2.0x	*16bit*	*0.25%*	*0.56%*	*1.79%*
	2.6x	*12bit*	*0.37%*	*1.25%*	*3.11%*
common video	2.0x	*16bit*	*0.24%*	*0.54%*	*1.78%*
	2.6x	*12bit*	*0.36%*	*1.24%*	*3.06%*
live video	2.0x	*16bit*	*0.24%*	*0.51%*	*1.72%*
	2.6x	*12bit*	*0.35%*	*1.22%*	*3.02%*

The aging phenomenon of OLED is presented in the Figures 6(a) ~6(c). Figures 6(d) ~6(f) show the compensation results of DBI with compression. The compression ratio is 2.0x, and the accumulative durations are 100h, 400h and 1000h, respectively. Figures 6(g) ~6(i) show the compensation results of DBI without compression.

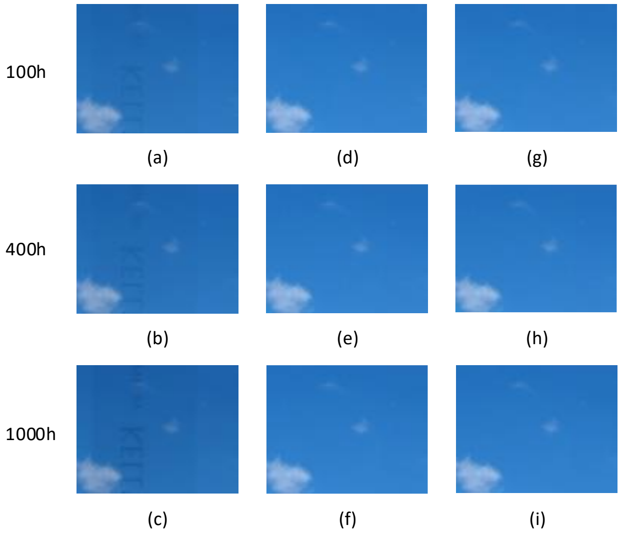

Figure 6. The compensation results.

From above table and figures, the compression performance of different data sets is similar, and there is no visible difference between the compensation results of DBI with compression and that of DBI without compression. It's clear that this method can save the storage space effectively while ensuring the performance of DBI.

4. Impact

In this paper, a specialized method has been presented to compress the DBI data:

- According to the characteristics of DBI data, set a smaller slice and a more suitable processing unit.
- In the process of quantization, adaptive dead zone is adopted to avoid unidirectional error accumulation.
- Because the DBI data range changes over time, adaptive Exponential-Golomb coding is used to improve compression performance.
- Since the distribution of the DBI data is non-uniform, RC mechanism is introduced to allocate bits.

The compression potential of DBI data is proved by experiments, and further optimization and experimental testing can be done on the basis of this method to save hardware resources.

5. References

1. Scholz S, Kondakov D, Lüssem B, Leo K. Degradation Mechanisms and Reactions in Organic Light-Emitting Devices. Chemical Reviews. 2015 Jul 31;115(16):8449–503.
2. Kim J, Ko J, Park J, Jeong G, Maeng H, Park S, et al. 35-4: A Compensation Algorithm for Degradation in AMOLED Displays. SID Symposium Digest of Technical Papers. 2018 May;49(1):445–8.
3. Zhao B, Huang R, Bu J, Lü Y, Wang Y, Ma F, et al. A new OLED SPICE model for pixel circuit simulation in OLED-on-silicon microdisplay design. Journal of Semiconductors. 2012 Jul;33(7):075007.
4. Mahmoud Al-Sa'di, Jaiser F, Sergey Bagnich, Unger T, Neher D. Electrical and optical simulations of a polymer-based phosphorescent organic light-emitting diode with high efficiency. Journal of Polymer Science Part B Polymer Physics [Internet]. 2012 Nov 15;50(22):1567–76. Available from: https://www.researchgate.net/publication/258670315_Electrical_and_optical_simulations_of_a_polymer-based_phosphorescent_organic_light-emitting_diode_with_high_efficiency
5. Jiang X, Xu C. Data-Counting Model for Empirical Prediction of OLED Degradation. IDW/AD '16 [Internet]. 2016 Dec 7; Available from: https://www.researchgate.net/publication/313163211_Data-Counting_Model_for_Empirical_Prediction_of_OLED_Degradation
6. bill. VESA Finalizes Requirements for Display Stream Compression Standard - VESA - Interface Standards for The Display Industry [Internet]. VESA - Interface Standards for The Display Industry. 2013 [cited 2025 Mar 6]. Available from: https://vesa.org/featured-articles/vesa-finalizes-requirements-for-display-stream-compression-standard/
7. Horé A, Ziou D. Image Quality Metrics: PSNR vs. SSIM [Internet]. IEEE Xplore. 2010. p. 2366–9. Available from: https://ieeexplore.ieee.org/document/5596999