A Novel Mura Compensation Algorithm for VR Displays

Jaechan Cho[1*], Jimin Seo[1*], Junyoung Park[1*], Yoonjong Yoo[1*], Byung Geun Jun[2*], and Ji Won Lee[1]

[1]LX Semicon, Inc., Seoul, Korea
[2]Meta, Inc., Sunnyvale, CA, USA

jccho@lxsemicon.com jmseo@lxsemicon.com parkjy4@lxsemicon.com yoonjong.yoo@lxsemicon.com byungpaul@meta.com petra.lee@lxsemicon.com

Abstract

The compensation of Mura defects is a key concern for enhancing image quality in display manufacturing. The Mura compensation method should be tailored to the specific characteristics of each application, such as automotive, notebook PCs, and virtual reality (VR) systems. VR applications, in particular, require compensation that accounts for additional luminance degradation and position adjustments due to lens shape, with area-power efficiency also being important. In this paper, we propose a Mura compensation algorithm that effectively improves luminance uniformity of VR displays. In addition, a compression method is applied, which reduces memory requirements while minimizing data loss, considering area-power efficiency. Our verification results on a test system with VR displays show that the proposed algorithm can achieve high luminance uniformity in the unique shape of VR displays and reduce implementation complexity with compressed compensation data.

Author Keywords

Mura Compensation; De-Mura; liquid crystal display; VR; Compression.

1. Introduction

In recent decades, liquid crystal displays (LCDs) have attracted significant interest and have become an essential part of the display industry [1]. During the manufacturing process of LCDs, detecting and compensating of Mura defects is a key challenge for improving image quality [2]. Mura defects are a kind of display defects that cause subpixel level non-uniformity and have irregular shapes [3]. In addition, the Mura defects have specific characteristics of each application, such as automotive, notebook PCs (NBPCs), and virtual reality (VR).

Among these applications, VR systems require compensation that accounts for additional luminance degradation and position adjustments caused by lens shape and foveated rendering (FR). This reduces the workloads of VR displays and mitigates human sickness by degrading image quality in the peripheral vision [4,5]. However, previous works with the Mura compensation did not consider these additional VR specific requirements, potentially leading to uniformity degradation [6,7]. In addition, several techniques utilizing machine learning, which can be applied to various situations, are not suitable for VR systems that require area-power efficiency [8,9].

This paper proposes a novel Mura compensation algorithm for VR displays based on the following factors: (1) By adjusting the compensation position and intensity according to the region, it effectively improves the luminance uniformity of VR Displays. (2) To reduce power consumption with low complexity, a compression method that considers the distribution of Mura compensation data, is applied. The design and implementation results of the Mura compensation processor for real-time processing are also presented.

2. Method

2.1 Mura Compensation for VR Displays

Fig. 1 shows the proposed Mura compensation system for VR displays. When the target display is powered on, a processing unit within the Mura station displays a series of pattern images, which are captured by the high-resolution camera system. Then, the encoding software processes these captured images to extract the compressed compensation data and stores it in the flash memory via the display driver integrated circuit (DDIC).

The Mura compensation hardware implemented in the DDIC can be categorized into two phases: 1) The calculation of the compensation data (referred to as De-Mura) and 2) the image compensation. The decoding logic is applied to the compressed compensation data stored in memory. When the sharing block size is large, the decoded data in block units is smoothed for each pixel using bi-linear interpolation. Then, a VR region compensation block adjusts the target position and intensity according to the region. To compensate for the uncaptured gray levels, the final compensation values are extracted using linear interpolation.

Since overcompensation can occur at low and high gray levels, data re-mapping and a convergence gain control are applied to the input image to prevent data saturation. Subsequently, a data compensation block generates a compensated image with improved luminance uniformity.

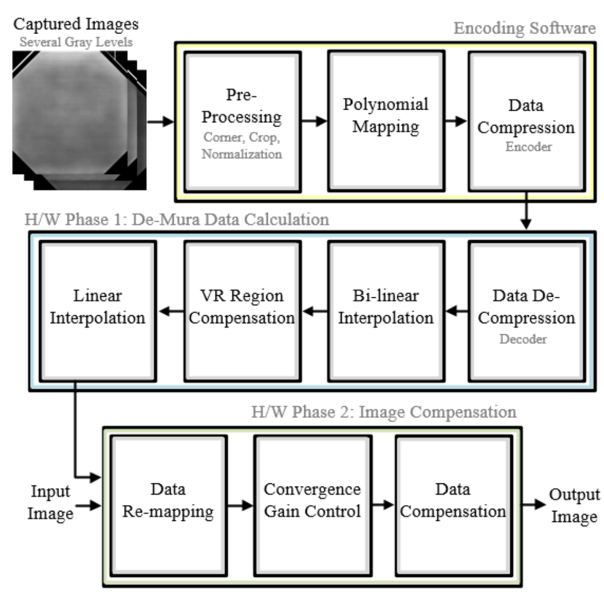


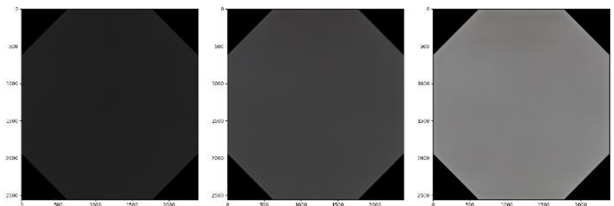
Figure 1. Schematic of the proposed Mura compensation system for VR displays

ISSN 0097-996X/25/5603-1551-$1.00 © 2025

2.2 VR Region Compensation

The VR display has a polygonal shape to be compatible with the human visual system (HVS) and the lens characteristics mounted on the device, as shown in Fig. 2. Therefore, the distribution of Mura data varies by region, depending on the starting point of the boundary and the location of the source driver IC. Since compensating the entire area with the same range affects uniformity and compression performance, the intensity of Mura compensation must be adjusted according to the area, and the physical compensation position should be considered accordingly.

The proposed VR region compensation adds an option to control the compensation range according to the region. Considering a typical VR display shape, it can be divided into up to three regions vertically, with each region compensated at different intensities ranging from x2 to x16. This approach improves uniformity by applying precise compensation for each area and increases compression efficiency by optimizing the distribution of the compensation data.



(a) 32 Gray levels (b) 64 Gray levels (c) 128 Gray levels

Figure 2. Example of the VR displays

2.3 Data Compression

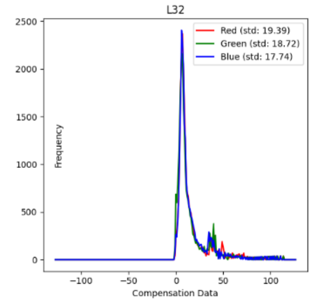
We utilized the proposed compression method for Mura compensation data to reduce power consumption with low complexity. Although it is based on simple differential pulse code modulation (DPCM), we employed a method that minimizes image quality degradation by analyzing the characteristics of Mura data in VR LCDs.

The captured Mura data is down-sampled to extract compensation values for each block size unit. Then, the encoding software calculates the difference between neighboring sub-pixels for compression. By analyzing the histogram of these differences, a codebook is created with representative values that have a high distribution. The size of the codebook depends on the number of compression bits allocated, which affects the compression ratio and reconstruction accuracy. Each differential value is then compressed by mapping it to the index of the closest value in the codebook.

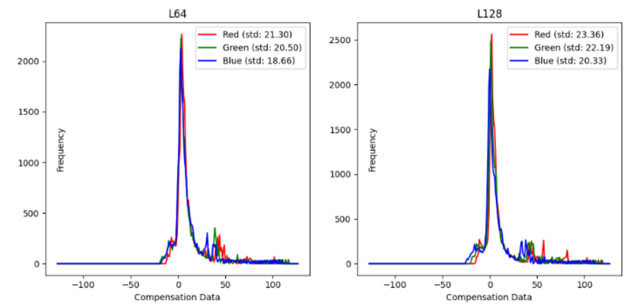
In addition, we applied a method to improve compression performance by utilizing the similar histogram characteristics of of R, G, and B compensation values in the VR LCDs, as shown in Fig. 3. Figs. 3(a) and (b) show the distribution of R, G, and B compensation values extracted from different gray levels, confirming that each exhibits a similar distribution in VR LCDs. Therefore, calculating the difference in compensation data between R/B and G is applied before the differential coding as follows:

$$\begin{cases} D_r(h,w) = R_c(h,w) - G_c(h,w) \\ D_b(h,w) = B_c(h,w) - G_c(h,w) \end{cases}, \quad (1)$$

where h, w are the horizontal and vertical indexes of the block, R_c, B_c, G_c are the compensation values of red, blue, green, and D_r, D_b are the target difference data.



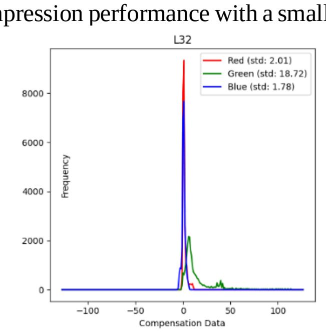
(a) 32 Gray levels



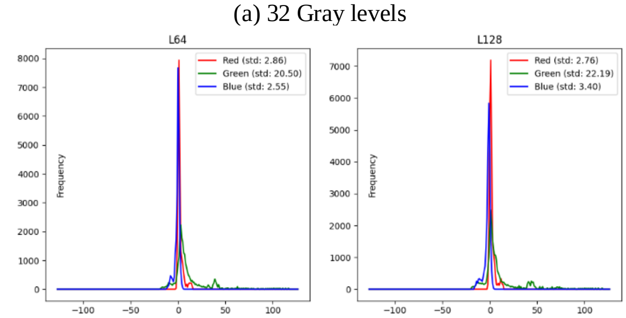
(b) 64 Gray levels (c) 128 Gray levels

Figure 3. Histogram of the Mura compensation data

Fig. 4 shows the histogram of D_r and D_b, and Table. 1 depicts the comparison of the standard deviations for each data set. At 32 gray levels (L32), the standard deviations of D_r and D_b were decreased by 89.63% and 90.49% compared to R_c and B_c, and they were reduced by 88.13% and 83.28% at 128 gray levels (L128), respectively. Similar reductions were observed at 64 gray levels (L64), with decreases of 86.57% and 86.33%. Therefore, the differences between neighboring sub-pixels of D_r and D_b and can be coded with fewer bits than the original R_c and B_c, improving compression performance with a smaller memory size.



(a) 32 Gray levels



(a) 64 Gray levels (b) 128 Gray levels

Figure 4. Histogram of the difference in compensation data between R/B and G (D_r and D_b)

Table 1. Comparison of the standard deviations for each data distribution

Gray Levels	Target data			
	R_c	D_r	B_c	D_b
L32	19.39	2.01	18.72	1.78
L64	21.30	2.86	18.66	2.55
L128	23.26	2.76	20.33	3.40

3. Experimental Results

3.1 Verification Platform

Fig. 5 depicts the FPGA test system used to verify the proposed Mura compensation. It consists of a main FPGA board with a MIPI daughter card, and a VR panel module. The MIPI daughter card receives a video data stream from a PC and transmits it to the proposed De-Mura processor in the FPGA programmable logic. The encoded Mura compensation data is stored in the SRAM, and the De-Mura processor reads and decodes the values. The compensated output image is then sent to the VR panel module.

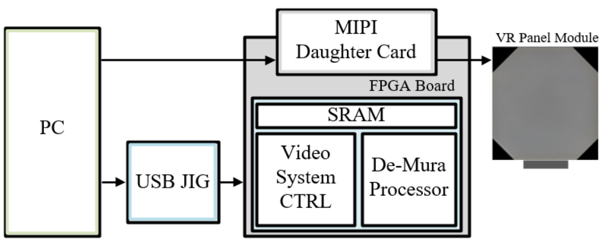


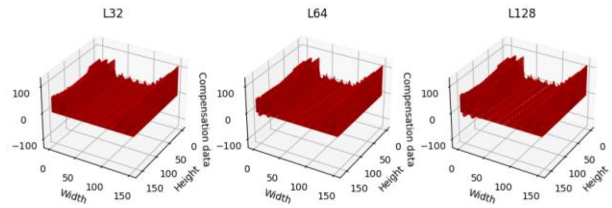
Figure 5. System diagram of the FPGA verification platform

3.2 Simulation Results

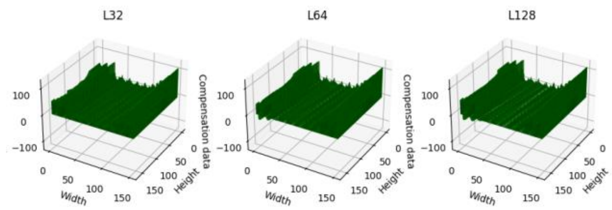
To evaluate the performance of VR region compensation, we captured 32, 64, and 128 gray levels for 50 VR panels on our verification platform. We extracted compensation values for these images, and set the compensation ranges of the upper and lower regions differently, as shown in Fig. 6, to enable adaptation to the characteristics of VR displays. Figs. 6(a)-(c) show the results of plotting compensation data by image coordinates for each gray level in R, G, and B, respectively. Since the distribution of compensation data is extracted for each area, the efficiency of the compression method based on the difference values between neighboring pixels in the horizontal direction is also improved.

Fig. 7 shows the evaluation results for the proposed compression method (C2) using D_r and D_b and the existing method compressed with R_c and B_c (C1), considering the compensation data features of the VR LCDs. The same compression ratio was applied, and the C1 method shows a red spectrum defect in the lower left and upper left corners, as shown in Fig. 7(b), while the proposed C2 method is properly reconstructed, as shown in Fig. 7(c). Since the data distribution in the corner area is generally different, when compression is applied to R_c and B_c data, the distribution appears wide as in Fig. 3, resulting in a large difference between the coding value and the actual value. However, the proposed C1 method can alleviate corner differences because it applies coding in a narrow distribution state as shown in Fig. 4.

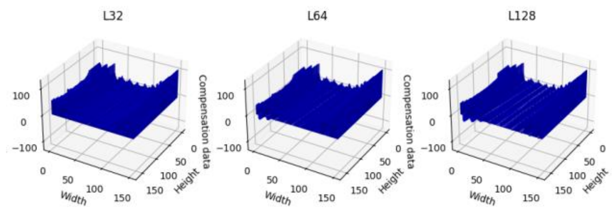
The peak signal-to-noise ratio (PSNR) performance, which measures the difference between the original and reconstructed compensation data, is presented in Table. 2. As a result, the proposed C2 method showed an average 29.94% improvement in PSNR performance compared to the C1 method. In addition, assuming the same PSNR performance, the proposed method can reduce memory requirements by applying a higher compression ratio than the C1 method.



(a) VR Region compensation for R_c

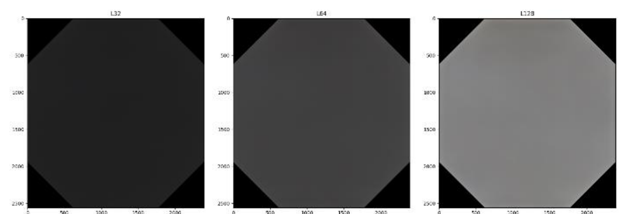


(b) VR Region compensation for G_c

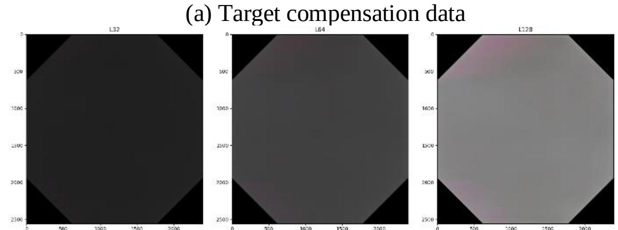


(c) VR Region compensation for B_c

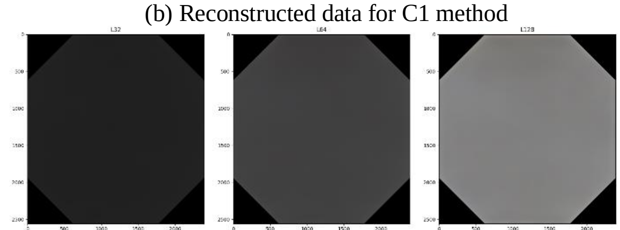
Figure 6. Example of the proposed VR region compensation



(a) Target compensation data



(b) Reconstructed data for C1 method



(c) Reconstructed data for C2 method

Figure 7. Comparison of the proposed compression method and previous results

Table 2. PSNR performance comparison by compression methods

Method	PSNR (dB)		
	L32	L64	L128
C1	54.05	43.74	34.80
C2	69.89	62.14	55.71

4. Conclusion

In this paper, we proposed the Mura compensation algorithm for VR displays. We utilized VR region compensation to improve the luminance uniformity of VR Displays and applied an efficient compression method that reduces memory requirements while minimizing data loss, considering area-power efficiency. The proposed De-Mura processor was implemented on an FPGA, and the test platform was constructed for verification in a display environment. Our verification results on a test system with VR displays show that the proposed algorithm can be applied to the unique shape of VR displays and achieve high compression performance.

5. References

[1] Jaechan C, Jongsang Y, Junyoung P, Ji Won L, and Sukju K, "Adaptive Local Backlight Dimming Control with Local Boosting," *SID 2024*, pp. 1092-1095, 2024.

[2] Ming. W, Zhang. S, Liu. X, Liu. K, Yuan. J, Xie. Z, Sun. P, and Guo. X, "Survey of Mura Defect Detection in Liquid Crystal Displays Based on Machine Vision," *MDPI Crystals*, vol. 11, no. 1444, pp.1-21, 2021.

[3] Weizhi L, Chang L, Qiang L, and Dahai Y, "Assigned MURA Defect Generation Based on Diffusion Model," *2023 IEEE Conf. on Computer Vision and Recognition Workshops (CVPRW)*, pp. 4395-4402, 2023.

[4] Lili W, Xuehuai S, and Yi L, "Foveated rendering: A state-of-the-art survey," *Computational Visual Media*, vol. 9, No. 2, pp. 195-228, 2023.

[5] Linghiu R, Naama A, and Jim Z et al, "Display and Optics Architecture for Meta's AR/VR Development," *IEEE Open Journal on Immersive Displays*, Vol. 1, pp. 71-78, 2024.

[6] Yao Z, Jianxu M, and Haoran T et al, "An Efficient Optical Mura Compensation System for Large Liquid-Crystal Display Panels," IEEE Trans. On Instrumentation and Measurement, Vol. 71., No. 5023213, 2022.

[7] Yini Z, Eunho L, and Haijun Q et al, "A Novel Compression Algorithm Application for De-Mura Compensation of AMOLED Panel," *SID 2023*, pp. 182-185, 2023.

[8] Chang-Hoon S, Chan-Yung K, Hyeon-Woon S, Ho-Min L, and Ji-Won L, "A Novel Compression Algorithm using Machine Learning for Mura Compensation of OLED Panel," *SID 2024*, pp.1593-1596, 2024.

[9] Zo-Han L, Qi-Yuan L, and Hung-Yuan L, "A Machine-Learning Strategy to Detect Mura Defects in a Low-Contrast Image by Piecewise Gamma Correction," *MDPI Sensors*, Vol. 24, No. 1484, pp. 1-18, 2024.