IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 14, 2018, accepted March 6, 2018, date of publication March 19, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2815765

ASIC Design for Real-Time One-Shot Correction of Optical Aberrations and Perspective Distortion in Microdisplay Systems

SUNG-WOOK EO[1,2], JOON GOO LEE[2], MIN-SEOK KIM[2], AND YOUNG-CHAI KO[1], (Senior Member, IEEE)

[1]School of Electrical Engineering, Korea University, Seoul 02841, South Korea
[2]Research and Development Group, RAONTECH, Seongnam 13591, South Korea

Corresponding author: Young-Chai Ko (koyc@korea.ac.kr)

This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No.10076638, One-chip Microdisplay with WVGA, WSVGA resolution module Development for Wearable Devices.

ABSTRACT As microdisplay systems have become more affordable for various applications, the need for larger field-of-view has become important, typically for near-to-eye (NTE) applications. However, for NTE applications, pincushion distortion and color aberration are difficult to avoid, due to the high magnification of the optics for the systems. In addition to optical distortion, various distortions such as perspective distortion may occur in combination. In this paper, we propose a novel architecture and algorithm for the real-time one-shot correction of compound distortion for practical use, without using traditional radial and tangential modeling. We confirmed the performance of our proposed scheme, which is implemented on an application-specific integrated circuit by testing with full high definition liquid crystal on silicon panels.

INDEX TERMS ASIC, distortion correction, microdisplay, optical distortion, perspective distortion, real-time system.

I. INTRODUCTION

The microdisplay system is becoming increasingly attractive in the mobile market, because it is smaller and lighter, and consumes less power than conventional displays. In particular, microdisplay systems play a major role in a variety of applications, such as head-mounted displays (HMDs), smart glasses, portable projectors, and head-up displays (HUDs). These applications are becoming more affordable for various users, since fast-moving displays and semiconductor technologies have reduced the price of display panels, while enhancing the picture quality of the panels in terms of the number of pixels and the color depth of each pixel. However, most of these systems require optics with high magnifying power to properly generate a display in front of a user's field of vision. This magnifying lens tends to distort the image on the display, and the use of a higher magnification lens to make a larger screen makes it more difficult to avoid distortion.

Optical aberrations are common phenomena for the near-to-eye (NTE) display system, due to the high magnification lens, as shown in Fig. 1. Therefore, sophisticated lens design may be needed to provide a larger field-of-view (FOV) while mitigating various optical distortions. Since it is very difficult to design near-perfect optics with low cost and small size for a given display system, many system designers choose moderate optics to trade-off the display quality against the system cost. If we design the display system with modest

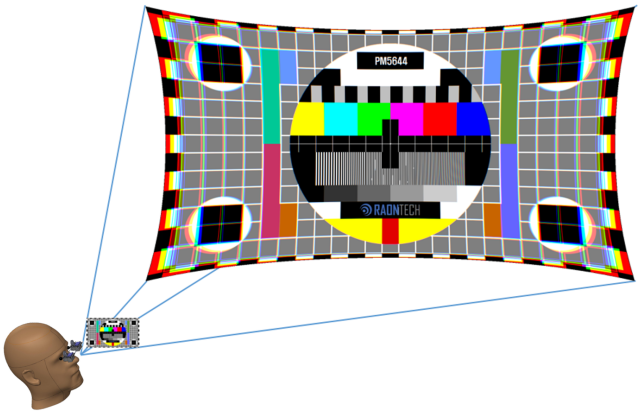


FIGURE 1. Optical aberration in the NTE display system.

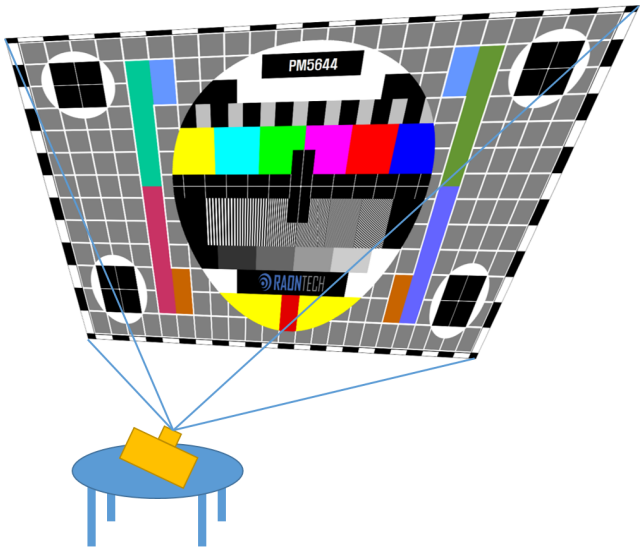


FIGURE 2. Perspective distortion in a projector system.

and imperfect optics for cost reduction, the optical aberrations become more severe, and the display would be unacceptable for end-users, due to the unbearably distorted result.

In a microdisplay system, display distortion as well as optical distortion may occur due to various factors. Fig. 2 shows that in projector applications, perspective distortion is caused by projection distance differences. This is the typical perspective distortion, in which a screen is created in a keystone shape by oblique projection on a flat wall. This type of distortion cannot be resolved, regardless of the quality of the optical design, and a different approach is required.

Many researchers have proposed correction methods to resolve the abovementioned optical or perspective distortion problems in other fields, such as the digital camera system [1], [2], cathode ray tube (CRT) display system [3], [4], and projector system [5], [6]. Amongst these diverse methods, [7] is worth mentioning, since it describes a method of warping images in real-time to cancel out complex geometric distortions caused by the lens system and tilted projection. The method uses a polynomial function for coordinate transformation, and the function is approximated from a generalized equation, which has fixed trigonometric terms. When comparing the original equation, the approximation is sufficiently simple to be calculated with hardware logics; thus, the proposed method could be implemented as an application-specific integrated circuit (ASIC). However, even the simplified equation is insufficient to adopt for mobile use, due to the complexity. A viable and realistic solution still needs to be found for the correction of complex optical and/or perspective distortions in the field of flat-panel display systems, while deliberating mobile use with low-power consumption and limited resources.

In this paper, we propose a novel architecture for the one-shot real-time correction of compound distortion, including optical and perspective distortions, in a microdisplay system, such as a liquid crystal on silicon (LCoS) panel or a digital micromirror device (DMD) display system. In our correction system, the active matrix of the display panel is divided into grids, and each rectangular-shaped section in the grid stores the number of distortions as a vector form for only four corners of the section. The correction system uses the vectors to generate artificially distorted frame images to compensate for the optical and/or perspective distortions in real-time. Since we do not use traditional radial and tangential modeling for the given optics or complicated numerical expressions for complex deformation, most calculations can be performed with shift/addition/subtraction and a small amount of simple multiplication, without the need for mathematical functions or division. This approach greatly reduces the design complexity, and facilitates the implementation of the proposed system in a hardware form in a pipeline manner. We implemented the correction logic with Verilog hardware description language (HDL) in a register transfer level (RTL) form, and verified the functional operation of the logic with a field programmable gate array (FPGA) [8]. We then successfully fabricated the logic into a commercial-grade display controller, which can simultaneously drive up to two LCoS display panels. The correction logic in the display controller has satisfactorily generated intermediate images from the input video stream for the corresponding display panels.

The remainder of this paper is organized as follows. Section 2 summarizes the background knowledge of microdisplay systems and the considerable distortions in the system. Section 3 presents the proposed architecture and implementation, while Section 4 describes the correction algorithm in detail. Sections 5 and 6 present the ASIC design and its test results, respectively. Section 7 concludes the paper, and discusses future works.

II. BACKGROUND KNOWLEDGE

For better understanding, we first briefly describe the microdisplay system, and the various possible distortions in the system.

A. MICRODISPLAY SYSTEM

A microdisplay is a miniaturized display that is designed for use with a magnification system [9]. Its physical screen size is generally less than 5 cm diagonal, and its pixel size is also very small, typically less than 10 μm, which is below visual acuity. Consequently, the full resolution is only discernible with optics for magnification. Fig. 3 shows an example of a microdisplay system that uses a LCoS panel and an optical system. Due to the invariably diminutive panel size, the application system size can easily be reduced for mobile use, despite the inevitable use of optics. The main applications using a microdisplay include video projection, such as pico-projectors, and HUD or NTE systems, such as HMD, camera viewfinders, and portable/wearable computers.

Nowadays, when we use a microdisplay system for versatile applications, a large FOV becomes more important for many reasons. NTE applications need a larger FOV for a more immersive experience. For projection-based devices,

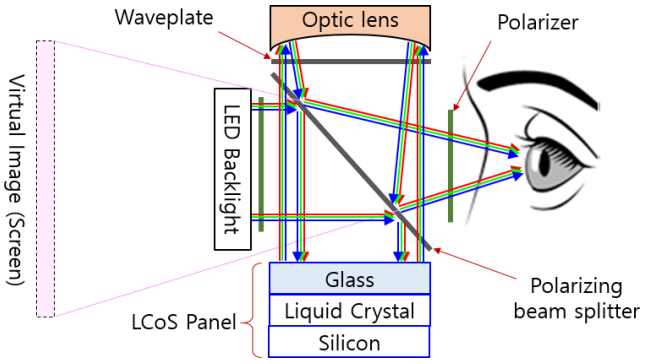


FIGURE 3. Example of a microdisplay system.

a larger screen with shorter projection distance is essential. However, increasing magnification for a larger FOV may result in distortion problems.

B. DISTORTIONS IN MICRODISPLAY SYSTEMS

Many types of distortions are involved in the practical use of a microdisplay system. The major distortions are optical distortion and perspective distortion.

1) OPTICAL DISTORTION

Optical distortion includes pincushion/barrel/mustache distortion and chromatic aberration due to the difference of light paths of imperfect optics. Optical distortion results in various types of deformation of images; thus straight lines can become bent, and curvy or wavy images and/or color dispersion can appear.

Pincushion distortion with chromatic aberration is common, especially in NTE applications, due to the magnification. Since the magnifying power of a typical magnifier tends to increase towards the lens periphery, the corners of images appear to be pulled outwards. It is also difficult for the lens to focus all the colors on the same convergence point, due to the different refractive indexes for light of different wavelengths. Fig. 1 exaggerates this phenomenon for better understanding. If we increase the FOV without precaution, the curvature will likely become more complex, and chromatic aberration will also become significant; thus, users may find the visual quality to be unbearable.

Many solutions to correct optical distortion have been proposed. Designing optics and optical systems with compensating elements may be one of these solutions. However, enhancing the lenses themselves is too expensive to minimize the distortion such that it becomes unnoticeable. Moreover, the designed optics can easily be bulkier and heavier, due to additional components.

An alternative method is to manipulate the image data for correction with digital signal processing. This can be performed with both software and hardware, while consuming additional computing power. The use of software has often been implemented by using a powerful desktop-level central processing unit (CPU) and/or graphics processing unit (GPU) [10]. Various open source libraries [11]–[13] and commercial tools provide a correction function for the post-processing of images and videos. In addition, Oculus Rift and some game consoles even support real-time correction with low-latency and high frame rate. Nonetheless, using a high-performance CPU/GPU consumes too much power and thus, it may not be a suitable solution for stand-alone wearable devices

Meanwhile, some systems use a sophisticated hardware logic that is designed only for the correction. This type of hardware is typically embedded in digital cameras or high-end digital camcorders to eliminate lens distortion [14]–[18]. The hardware logic processes image data by using massive parallelism and pipelining. Even though many proposals have been presented on methods to implement the hardware, most of these are not suitable for display purposes because of the lack of real-time process capability for high-definition and deep color depth, whilst considering low latency, high refresh rate, and low power consumption.

Nonlinearity caused by the spherical property of a lens renders it difficult to compensate for optical distortions in the digital process. Mathematically, this distortion is well explained by radial and tangential modeling [19], and it is basically high order polynomials. This distortion can be corrected using the Brown-Conrady model by warping the input image with a reverse distortion. Although this model corrects the optical distortion caused by a lens, a correction logic still needs memory access and an inherent number of complex calculations. Consequently, previous works have primarily focused on calculation reduction. Some of the approaches are categorized as follows.

(A) Look-up table (LUT)-based processing uses precalculated values stored in LUTs, which are referenced at run-time for given optics for a specific display [20]–[22]. This may be applicable for a low-resolution display system, since as the number of pixels that are processed increases, the size of LUT needed also increases. (B) The Coordinate rotation digital computer (CORDIC) may be used for image processing due to its simple and efficient algorithm to calculate trigonometric functions [23]–[25]. CORDIC only requires addition, subtraction, bitshift, and table lookup for the calculation of trigonometric, square-root, division, and many other computing tasks. Generally, CORDIC needs deep-pipeline in practical implementation. (C) A linear approximation scheme can be used for distortion correction. A display is divided into a grid of many rectangular-shaped sections, and each section uses a differently approximated linear function derived from a specific lens model. A high-definition image may need a denser grid to make a smoother output image.

2) PERSPECTIVE DISTORTION

Perspective distortion is also one of the major distortions due to perspective. This distortion can generally be seen in applications for multimedia projection, when projecting an image onto a two- or three-dimensional (2- or 3D) projection surface. If the multimedia projector is not placed perpendicular to the horizontal/vertical centerline of the screen,

the projected view deforms the output picture on the screen. In other words, tilted projection tends to make a keystone-like distorted image on the projection surface, as shown in Fig. 2.

Perspective distortion can also be corrected by manipulating image data with digital signal processing, while risking degradation of picture quality. The simple vertical keystone effect is common for most projector applications, and is fairly easy to correct [26]. The basic correction method for the effect is line-by-line horizontal scaling with different scaling ratios. Since this requires only two line buffers and straightforward computation, most projectors provide a digital vertical keystone correction function. The horizontal keystone is more difficult to compensate than the vertical case. This is because vertical scaling for horizontal keystone correction uses the pixel values of other lines, and it needs more complex memory access. Similarly, it is considerably more difficult to cancel out a 2D complex keystone; thus, this correction function has only been available in high-end projectors. However, as mobile projectors have become more popular, the function is also required for low-cost projectors. Moreover, automatic correction is considered for mobile devices, by using a camera, tilt sensor, and gyroscope sensor.

3) COMPOUND DISTORTION

Some applications experience much more complicated distortion with both optical and perspective distortions. For example, tilted projection onto a concave screen, such as a windshield, is common for autonomous HUD application, and shows the complex shape of perspective distortion [27]. Moreover, a windshield is also an optical component, which is likely to add various optical aberrations. This type of deformation of view is difficult to model, and thus it is difficult to compensate properly.

As these distortions are difficult to control simultaneously, the design goal is to reduce distortions to negligible levels, rather than to make the perfect view.

III. ARCHITECTURE

Nowadays, most microdisplay systems use a device called display controller to create an optimal screen view that is customized for the different features of various backplane designs. The display controller processes video signals received from video source such as application processor (AP), GPU, personal computer (PC), and so on, through a standard video interface. It converts the signal to a suitable format according to the corresponding panel interface, and transfers the re-arranged video data to the panel(s). Finally, the panel expresses the inputted pixel values on a pixel matrix.

For high performance display systems, the display controller tends to embed some image processing modules to improve the picture quality by providing panel specific calibration. However, some of the image processing techniques require numerous complex calculations and hardware resources, such as memory, to implement them as a hardware logic. In other words, a computing intensive or memory intensive algorithm is not easy to realize. Unlike simple algorithms, such as brightness or contrast control algorithms, the predistortion of the display for optical and perspective deformation correction is in general both computing and memory intensive, so it is difficult to implement in display controller. Until now, in some NTE application systems suffering from severe optical distortion, a display controller only received the predistorted video data generated by another computing machine for the distortion correction. However, in the proposed system, the distortion compensation is performed in real time in the display controller.

In general, a microdisplay system using field sequential color (FSC) needs a frame memory that temporarily stores at least one entire image frame color-by-color to show the proper picture. It also requires a much higher refresh rate for panel operation than the input frame rate, in order to suppress flicker and color breakup [28]. For this reason, every video data processing module after the frame buffer needs more computing power than the modules before the buffer. In order to perform timing-critical tasks without stuttering, an increment of the operational clock frequency or data bus widening is required when designing.

Consequently, the position of the distortion correction module becomes very important, since it greatly affects the complexity and size of the design. If the position of the distortion correction module is ahead of the frame buffer, the calculation of each pixel for predistortion per one clock cycle is relatively simple. However, the size of line buffers needed for distortion correction would be considerably large, since all red/green/blue (RGB) data must be stored in the line buffers for proper processing. If the correction module uses the frame buffer directly for random access to remove line buffers, the memory bandwidth of the frame buffer would significantly increase, and this negatively affects the overall performance and power consumption due to memory bus congestion. In contrast, if the module resides after the frame buffer, designing the module becomes more complicated. Increasing the clock frequency reduces the timing margin for the processing module. Using a wider bus, i.e. more data channels, would need multiple pixel positions, and makes it difficult to calculate the referencing positions of several consecutive pixels per one clock cycle. However, only one color field dataset is necessary for the correction process, and thus the size of the line buffers can be reduced to one-third. In our display controller architecture, it is difficult to share the line memory of distortion correction module with other functional blocks. Because most functional blocks operate independently, some blocks must wait to access the memory when multiple blocks are activated. It is not suitable for a system that processes a real-time image at a high frame rate to drive a microdisplay panel. Also, in general, the area of the line buffers is larger than the logic for distortion correction, and the area of the design is directly related to the cost of an integrated circuit (IC). Therefore, even if the design is more complicated, we place the distortion correction module behind the frame buffer, as shown in Fig. 4.

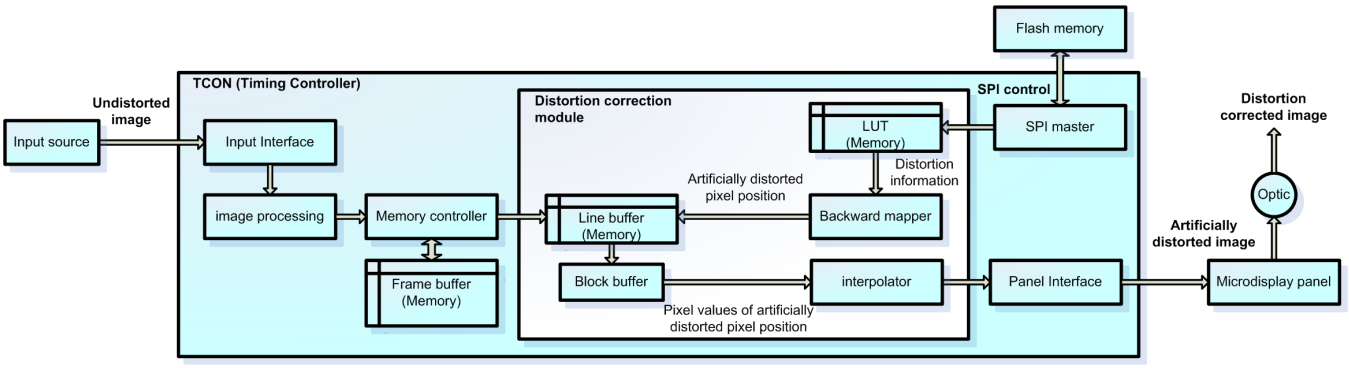


FIGURE 4. Proposed architecture for distortion correction in a microdisplay system.

The proposed distortion correction module consists of LUTs, a backward mapper, line buffers, block buffers, and an interpolator. The LUT stores the distortion information at the representative position of the screen. Three LUTs are used to correct chromatic aberration, and each LUT stores distortion information corresponding to each color field of RGB. The backward mapper uses the distortion information received from the LUT, and then computes the referencing coordination for each pixel to be artificially distorted, by using a bilinear approximation method. Based on the calculated pixel position, the appropriate data value is obtained from the line buffers. The vertical correction capability of the distortion correction module is determined by the number of lines able to be held by the line buffers. As mentioned above, since it is necessary to calculate the referencing pixel positions and the values of several pixels in one clock cycle, small block buffers are used to access multiple pixel values at the same time. Unlike other buffers that use memory, such as line buffers and frame buffers, it is better to use flip-flops for free access. An interpolator is used to produce a smoother image. After investigation of various interpolation algorithms, we decided to use a bilinear interpolator to reduce the complexity. Using a more powerful interpolator will result in a slightly sharper image, but it tends to require much more complex design, and the size of the block buffer can increase.

In some applications, the distortion information corresponding to the entire pixel is stored in the LUT. In this case, memory size larger than the frame buffer is needed to store the distortion information. For example, when setting the display resolution to full high definition (FHD), the distortion correction capability of the column direction is from -10% to 10% of the number of column pixels, and the distortion correction ability of the row direction is from -10% to 10% of the row pixel number. In this case, the number of bits of the column and row direction distortion information is 9 and 8 bits, respectively, and thus 17 bits are required for one pixel. To compensate for different distortions by RGB color, 51 bits per pixel are required. The distortion information of one pixel is 2.1 times larger than the one-pixel video data of the RGB 888 format. That is, a memory of 2.1 times greater than the frame buffer should be used to store the distortion information. Of course, this depends on the target resolution and distortion correction capability, but it also requires a memory as large as the frame buffer. Since the microdisplay system is mostly used for mobile applications, it is very important to reduce the area and power of the display controller; this method is therefore not suitable. In our proposal, only the distortion information at the representative point is stored in the LUT, and the backward mapper calculates the amount of distortion at the intermediate position using these values.

The distortion information stored in the LUT may be obtained from an optic manufacturer, or may be calculated mathematically. If it is difficult to calculate the value, due to various complex distortions other than simple radial and/or tangent distortion, it is possible to experimentally find the most appropriate value while changing the LUT. In this case, if the display controller provides a proper pattern generator for the experiment, the appropriate LUT may be found more easily.

In some applications, such as the HMD, where the projection angle and optics remain unchanged, the LUT is hardly changed after the entire configuration, including the optical system, is completed. On the other hand, in applications such as pico projectors where the projection position changes frequently or in developmental stages, the LUT needs to be updated. However, it is not easy to update the LUT value rapidly for the changed distortion environment. Additional processing units are required to obtain mathematical calculations, or considerable time is required to obtain them experimentally. To solve this problem, we can prepare several LUTs that are expected to be used frequently in an application system, and store the LUTs in a flash memory. Then, when the distortion environment changes, the LUT that can compensate for the specific distortion is loaded from the flash memory, and then used.

IV. CORRECTION ALGORITHM

The distortion correction module produces artificially distorted images in the opposite direction of the display deformation. Formal distortions, such as radial and tangential

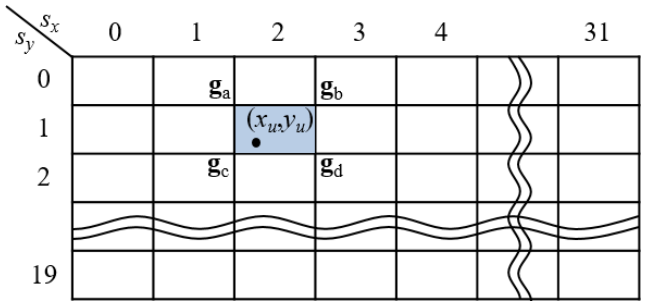


FIGURE 5. Example of addressing pixel position and the corresponding sector of a grid.

optical distortion, can be easily corrected by mathematical modeling. However, in applications using microdisplay, complex distortions often occur in addition to formal distortions, and it is not easy to correct them with a traditional equation model. We propose an algorithm that is simple to reduce the complexity of a hardware design, and can compensate these unexpected complex distortions simultaneously. Our goal is not to perfectly compensate for the distortion, but to greatly reduce the distortion or ensure that the end user hardly notices a distorted view.

The proposed correction algorithm uses a type of bilinear approximation to calculate the amount of distortion at every pixel position, by using the distortion information at representative points. The active matrix of the display panel is divided into a lattice pattern, and the positions of vertexes in each rectangular region become the representative points. The distortion information at these positions is stored in the LUT. As the number of representative points increases, the distortion correction becomes more natural; however, this requires more memory for the LUT and increases the LUT update time. Therefore, it is important to balance image quality against hardware resources.

The following steps describe how the correction algorithm generates an inversely distorted image for distortion correction. In general, the sequence of processing for each pixel is from top to bottom for row position, and left to right for column position.

1. Find the most appropriate sector (s_x, s_y) for the pixel position addressed (x_u, y_u) in the specified grid configuration. This is an intuitive process. For the FHD panel example of the 32 by 18 grid, the pixel (x_u, y_u) within $(120, 60)$ and $(180, 120)$ corresponds to the section $(s_x, s_y) = (2, 1)$, as shown in Fig. 5.
2. Load four vector values for the edges of the selected section from the LUT. These values represent the distance of the difference between the original pixel position of each edge, and the position to compensate for misalignment due to distortion:

$$\begin{aligned}
\mathbf{g}_a &= (g_{a,x}, g_{a,y}) = LUT\,(s_x, s_y)\\
\mathbf{g}_b &= (g_{b,x}, g_{b,y}) = LUT\,(s_x + 1, s_y)\\
\mathbf{g}_c &= (g_{c,x}, g_{c,y}) = LUT\,(s_x, s_y + 1)\\
\mathbf{g}_d &= (g_{d,x}, g_{d,y}) = LUT\,(s_x + 1, s_y + 1)
\end{aligned} \tag{1}$$

$\mathbf{g}_a$, $\mathbf{g}_b$, $\mathbf{g}_c$, and $\mathbf{g}_d$ are vector values in the top left, top right, bottom left, and bottom right, respectively, of the four corners.

3. Calculate the relative coordinates (x_r, y_r) for each (x_u, y_u) within the sector (s_x, s_y). The origin is the top left corner.
4. Calculate the amount of distortion at each pixel position, using the distortion values at the four corners and relative coordinates. Any interpolation scheme can be applied for this step; for simplicity, we use bilinear interpolation:

$$\begin{aligned}
(x_{d,ac}, y_{d,ac}) &= (\mathbf{g}_c - \mathbf{g}_a)\, y_r h_s + \mathbf{g}_a\\
(x_{d,bd}, y_{d,bd}) &= (\mathbf{g}_d - \mathbf{g}_b)\, y_r h_s + \mathbf{g}_b\\
(x_d, y_d) &= (x_{d,ac}, y_{d,ac}) - (x_{d,bd}, y_{d,bd})\, x_r v_s\\
&\quad + (x_{d,ac}, y_{d,ac})
\end{aligned} \tag{2}$$

where, h_s and v_s are precalculated values that are the reciprocal of the horizontal and vertical unit sizes for each section, respectively. If we need to support constraints, such as the resolution and the size of the grid, we can easily reduce the burden on the calculator by eliminating many multipliers for Eq. (2), since the h_s and v_s are constants.

5. Determine artificially distorted pixel coordinates corresponding to each undistorted pixel:

$$(x_a, y_a) = (x_u, y_u) + (x_d, y_d) \tag{3}$$

6. Calculate the fractional part for interpolation processing. Each pixel position calculated by Eq. (3) is a rational number. To reduce hardware complexity, we use fixed-point arithmetic. As the number of bits in the fractional part increases, the output image becomes smoother.

$$\begin{aligned}
x_{a,f} &= x_a - \lfloor x_a \rfloor\\
y_{a,f} &= y_a - \lfloor y_a \rfloor\\
x_{a,i} &= x_a - x_{a,f}\\
y_{a,i} &= y_a - y_{a,f}
\end{aligned} \tag{4}$$

where x_f denotes the fractional part of x and x_i denotes the integer part of x.

7. Read the neighboring pixel values $\mathrm{P}(x_{a,i}, y_{a,i})$, $\mathrm{P}(x_{a,i} + 1, y_{a,i})$, $\mathrm{P}(x_{a,i}, y_{a,i} + 1)$, and $\mathrm{P}(x_{a,i} + 1, y_{a,i} + 1)$ from line buffers for interpolation. $\mathrm{P}(x, y)$ denotes the pixel value of a given pixel coordinate (x, y).
8. Interpolate pixel values using the fractional parts in step 6 and neighboring pixel values in step 7. We use bilinear interpolation:

$$\begin{aligned}
&(1 - x_{a,f})(1 - y_{a,f})\,\mathrm{P}(x_{a,i}, y_{a,i})\\
&\quad + (1 - x_{a,f})\, y_{a,f}\,\mathrm{P}(x_{a,i}, y_{a,i} + 1)\\
&\quad + x_{a,f}(1 - y_{a,f})\,\mathrm{P}(x_{a,i} + 1, y_{a,i})\\
&\quad + x_{a,f} y_{a,f}\,\mathrm{P}(x_{a,i} + 1, y_{a,i} + 1)
\end{aligned} \tag{5}$$

V. ASIC DESIGN

The previous sections provided general information about our proposed architecture and algorithms. In this section, we provide more specific details on how our ASIC is designed.

Before starting the design, the design target for the distortion correction capability should be determined. Our main target is to restore 10% TV distortion in FHD resolution and 15% TV distortion in high definition (HD) resolution to a total of 0% TV distortion. In the case of a more severe distortion, the distortion is mitigated and a more natural image is given to the user than a nondistortion corrected image, but it cannot be completely corrected. The TV distortion value is defined as the ratio to the height difference between the corner of the image and the center [29].

$$\text{TV Distortion}(\%) = \frac{corner\ height - cneter\ height}{center\ height} * 100$$

According to this definition, our target can be realized by predistorting the image from -54 to $+54$ lines in the vertical direction. Of course, this definition cannot be directly applied to the range of predistortion correction, since the TV distortion value is obtained by measuring the height of the image after being distorted by the optical. However, in general, distortion tends to increase toward the edge. By predistortion, the vertices become closer to the center, so that even distortion larger than the correction range can be corrected.

The second target for determining the range of distortion correction in the horizontal direction is the keystone correction capability. Our distortion correction module will cover 25% vertical keystone at FHD resolution and 40% vertical keystone at HD resolution. This keystone distortion value is defined as the ratio of the difference between the top width and the bottom width [5].

$$\text{Keystone}\ (\%) = \left(1 - \frac{Top\ width}{Bottom\ width}\right) * 100$$

We set the distortion correction range in the horizontal direction from -256 to 255 to achieve the target in HD resolution, and we compensate up to 27.6% of the vertical keystone in the FHD resolution, which is larger than the target.

Finally, the distortion correction range in the vertical and horizontal directions is determined, and more distortion can be corrected in the horizontal direction than in the vertical direction. This is a reasonable design specification considering the type of distortion and the design cost. In the case of optical distortion, the distortion is larger in the horizontal direction. Because distortion is worse at a position far from the center, and we mainly support the wide resolution images. Keystone distortion is mainly observed in mobile projector applications, which are typically used to project upwards rather than perpendicular to the screen. As a result, vertical keystone distortion occurs and a predistortion in the horizontal direction is required to correct this. In terms of the design cost, the vertical distortion correction range not only affects the number of line buffers occupying the largest area, but also affects the delay time due to distortion correction. Therefore, it is advantageous to set the vertical distortion correction range as small as possible in comparison with the horizontal distortion correction range.

Based on the determined distortion correction range, we assigned 9 bits for the horizontal distortion correction range and 7 bits for the vertical distortion correction range. That is, the finally determined horizontal distortion correction range is -256 to $+255$, and the vertical distortion correction range is -64 to $+63$. We can therefore compensate 11.9% distortion at FHD resolution and 17.8% distortion at HD resolution, which means that we can restore distortion to be slightly more than our target. Also, the distortion correction value at one representative point can be represented by a total of 16 bits; i.e. 2 bytes are expressed in units of bytes, so that data can be efficiently processed. For example, the memory for updating the LUT and the update time can be efficiently used.

The delay of distortion correction depends on the number of line memories and the processing time of one line. The memory required to cover our vertical distortion correction range is 128 line memories. The resulting delay corresponds to 64 line times, which is half of the memory used. One-line time is calculated using the horizontal resolution, the number of channels, the synchronization signal time, and the clock frequency. In the case of FHD resolution, we used 8 data bus channels. That is, the number of time cycles required in the data field per line is 1920/8, and in our packet structure, two additional clocks are required as synchronization signals. With a 125 MHz operating frequency for the module, the total latency caused by the module is about 124 us, which is imperceptible by the user.

$$((1920/8 + 2)/125\text{MHz}) * 64 = 123.90\ \text{us}$$

We limited some parameters to reduce the circuit complexity of the calibration algorithm. Our ASIC supports mainly HD, FHD, and quad high definition (QHD) resolutions and additionally 2K1K, 2K2K, and 2HD resolution. Considering that the aspect ratio of the mainly supporting resolution is 16: 9, we decided that the number of grids also has a ratio of 16:9. As the number of grids increases, the predistortion becomes smoother, but it requires more memory, and the time to update the LUT increases.

Therefore, the number of grids should be as small as possible while also allowing natural predistortion. We have simulated various distortions to find the appropriate number of grids. Since the distortion between adjacent pixels does not change rapidly in most distortion environments, a small number of grids is sufficient. In the simulation, the worst distortion environment was side-by-side 3D. Fig. 6 shows a simulation image of the predistortion at various grid numbers for side-by-side 3D images. The number of grids used in Figs. 6(a), 6(c), and 6(e) is 16 by 9, 32 by 18, and 48 by 27, respectively. While the overall predistortion images shown in Figs. 6(a), 6(c), and 6(e) seem to be the same, when enlarged, an unnatural part is recognized when the number

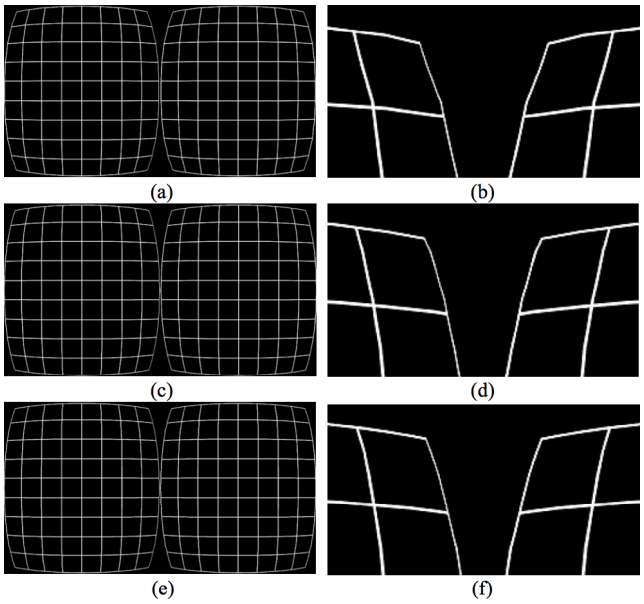


FIGURE 6. Simulation result image for side-by-side 3D predistortion when grid size is (a) 16 by 9, (c) 32 by 18, and (e) 48 by 27. (b), (d), and (f) are magnified images of the upper center of (a), (c), and (e), respectively.

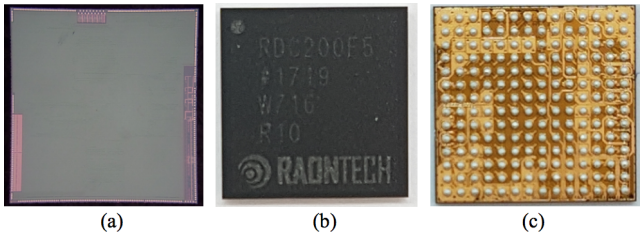


FIGURE 7. Photo of our ASIC (a) die, (b) package top, and (c) package bottom.

of grids is small. Figs. 6(b), 6(d), and 6(f) are magnified images of the upper center of Figs. 6(a), 6(c), and 6(e), respectively. In Fig. 6(b), unnatural lines are observed at the corners, whereas in Figs. 6(d) and 6(f), smooth lines are shown. Thus, the number of grids is determined by 32 by 18, which is the smallest number, but natural predistortion is possible. Considering the number of grids and the number of bits allocated to the distortion correction value, the LUT size for one primary color is about 10 kbit, and total size of the LUTs for RGB colors is about 30 kbit.

With this design specification, we designed the display controller with a distortion correction module, and fabricated it with TSMC 55nm technology. Fig. 7 shows the die photo, package top, and package bottom of our display controller ASIC. The specifications and design values for our chip are summarized in TABLE 1. The maximum operating frequency of the distortion correction module is 140 MHZ. The gate count of each element in this table depends on the synthesis report. Note that each line buffer can hold up to 2,560 pixels for QHD resolution support. To further reduce the size of the line buffers, we used a 2-port register file,

TABLE 1. Specifications and design values of distortion correction module in our display controller ASIC.

Resource type	Cost of the design
Optical distortion correction	11.9% TV distortion in FHD 17.8% TV distortion in HD
Vertical keystone correction	26.7% keystone in FHD 40.0% keystone in HD
Range of distortion correction	-64 ~ +63 for vertical direction -256 ~ +255 for horizontal direction
Memory for line buffer	2560 x 8 bits (2-port register file) x 128
Latency	About 124 us
Supporting resolution	HD, FHD, QHD, 2K1K, 2K2K, 2HD
The number of grid	32 x 18
LUTs size	About 30 kbits
Technology	TSMC 55nm Low power
Operating frequency	Max. 140 MHz
Gate count of entire chip	21.3 MGE [a]
Gate count of line buffer	3.79 MGE
Gate count of block buffer	0.05 MGE
Gate count of LUTs	0.03 MGE
Gate count of logic	0.72 MGE

[a]MGE means Mega gate equivalents

instead of a dual port static random access memory (SRAM), for each memory.

TABLE 1 shows that the area occupied by the line buffers is a very large portion of 17.8% of the entire chip. If the distortion correction module is located in front of the frame buffer, the size of the line buffer will be three times larger because three color data must be stored. Instead, if the module is located after the frame buffer, 8-pixel data in one primary color will be processed per one clock, but if the module is located before the frame buffer, one pixel data in three colors will be processed per one clock. Therefore, some of the logic will be reduced by 3/8. By simply estimating the size, the gate count of the line buffer is increased from 3.79 MGE to 11.37 MGE, and the gate count of logic is reduced from 0.72 MGE to 0.27 MGE, and finally the gate count of the entire chip is increased by 33% from 21.3 MGE to 28.43 MGE. Note that the gate count of the block buffer and LUTs is negligible because it is very small. Therefore, in FSC display controller, it is reasonable to place the distortion correction module after the frame buffer, even if the logic of the distortion correction module becomes more complicated, as described in Section III.

VI. DESIGN VERIFICATION AND PERFORMANCE EVALUATION

Fig. 8 shows the test environment of a microdisplay system we established to verify the proposed distortion correction architecture and algorithm. The system consists of an evaluation board with a fabricated display controller IC, and two display panels for a binocular application.

The evaluation board includes the display controller ASIC, power supplying circuits, a serial peripheral interface (SPI) flash memory to store the boot program binary and LUTs, and an HDMI bridge IC. The HDMI chip receives HDMI video

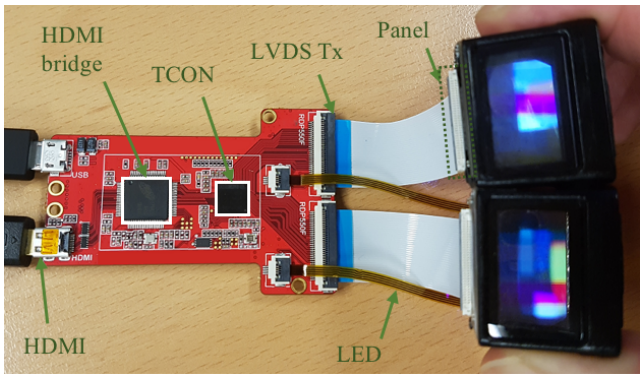


FIGURE 8. Test environment of the microdisplay system with binocular HMD.

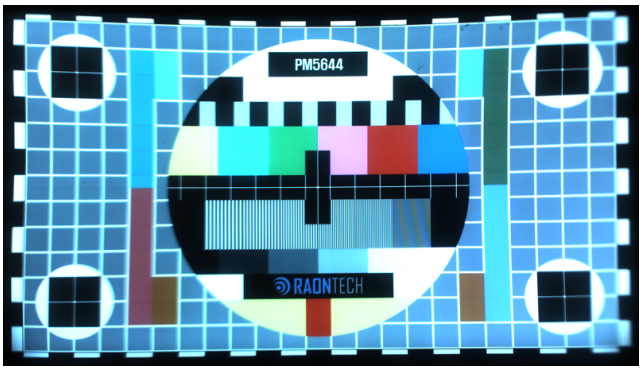


FIGURE 9. Pincushion distorted screen by optical system in actual microdisplay system.

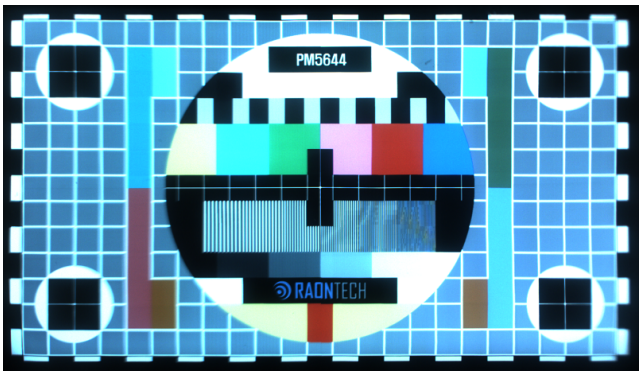


FIGURE 10. Corrected view, after applying barrel-type inverse distortion to remove the pincushion distortion.

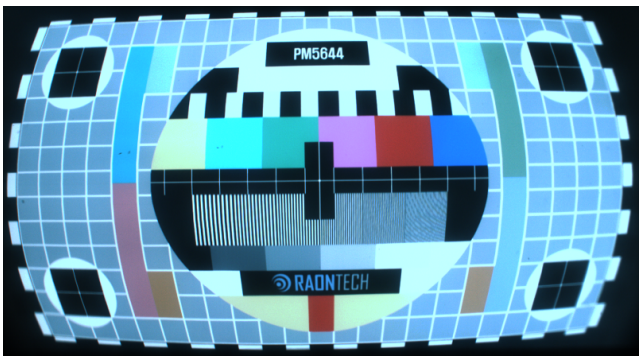


FIGURE 11. Example view to compensate for more severe pincushion distortion.

signals from other systems, such as notebooks or multimedia players, then converts the signals to the appropriate video format for the display controller, and delivers the formatted video data to the display controller. In the display controller, the transferred data are stored in the frame buffer memory, and converted into the FSC format. If the correction module is enabled, each field data is manipulated to form a predistorted image in the distortion correction module, and is then transmitted to the panels. The interface between display controller and the panels is a low-voltage differential signaling (LVDS) interface, for simple and fast transmission.

The used FSC LCoS panel displays consecutive frame images color-by-color with an optical system and an LED backlight unit. Each panel is capable of displaying up to FHD or 2K1K resolution. The diagonal size of the panel is 0.55 inches, and the optics is designed to produce a virtual screen of approximately 65 inches at a distance of 2 meters; i.e. the FOV is about 40 degrees.

Fig. 9 shows an image magnified by the optics without distortion correction. Although the optics is designed to eliminate as much optical distortion as possible, noticeable pincushion distortion is observed, especially at the edges of the screen, because it greatly magnifies for large FOV. A complimentary metal-oxide semiconductor (CMOS) camera [30] with a lens [31] is used to capture the screen of the panel, and the optical aberration caused by the lens itself is negligible (less than 0.1% TV distortion). Therefore, the captured image does not differ from the viewer's perception. Fig. 9 shows a standard mobile imaging architecture TV distortion value of about 2.8%. Fig. 10 shows an image with optical distortion correction, and the TV distortion value is 0%, which is perfectly corrected. The distortion correction module uses distortion information to predistort the image into a barrel, which is the inverse distortion of the pincushion, and transmits it to the panel. This predistorted image passes through the optics, and appears as a distortion-free image.

Fig. 11 shows extreme barrel predistortion in the same experiment environment as that above. This makes it possible to use higher magnification optics, even if the optics generates more severe pincushion distortion. Therefore, it will be useful for applications such as virtual reality (VR) or ultra-short throw projectors that require very large FOVs.

It is not easy to design a compact optical system with a large FOV without chromatic aberration. However, by performing the chromatic aberration correction using our correction module as shown in Fig. 12, the burden on the optical system design for the large FOV can be reduced. In addition, as described previously, very few additional hardware resources are required to perform chromatic aberration correction in our architecture. Only additional LUTs are needed for each primary color, and other logic and memory

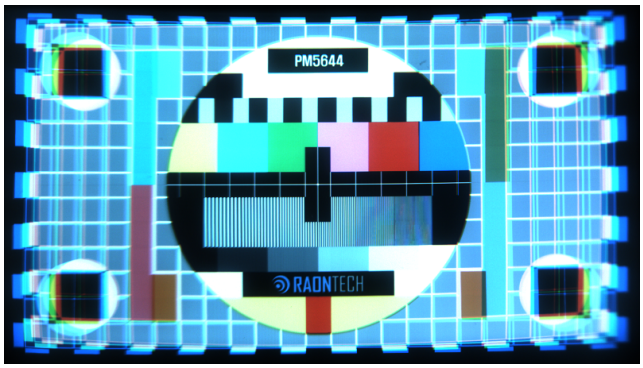


FIGURE 12. Example view to compensate for chromatic aberration.

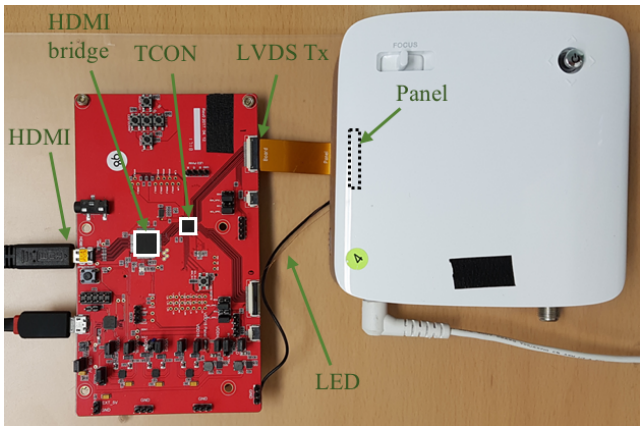


FIGURE 13. Another test environment of the microdisplay system that verifies keystone correction with a mobile projector.

FIGURE 14. Keystone distortion caused by tilting the projector 20 degrees upward from the ground.

FIGURE 15. Keystone correction in the horizontal direction in the distortion shown in Fig. 14.

are shared. The size of the memory used in one LUT is 0.01 MGE, corresponding to 0.05% of the total distortion correction module.

In the HMD system, the keystone distortion may occur when the reflection angle of the optical system is adjusted for various purposes such as reducing the thickness of the optical system. However, keystone distortion occurs more frequently in mobile projectors. For keystone verification, we set up another verification environment using a mobile projector as shown in Fig. 13. The evaluation board used in this verification environment is an initial version of various verifications. The overall circuit configuration is similar to that shown in Fig. 8, and only the specifics differ. The power supply circuit is subdivided to verify the power characteristics, and the button is used as a user input interface.

A generic mobile projector that is readily available on the market is used, and only a microdisplay panel is replaced with another panel that is compatible with our display controller output interface. The 0.55-inch panel used in the previous test environment is too large to use for this test because the original panel used in the mobile projector is 0.31 inches. Therefore, we replaced it with a 0.36-inch panel that is compatible with our display controller and mobile projectors. The 0.36-inch panel has the same HD resolution as the original 0.31-inch panel and the FOV of the mobile projector optics is 35 degrees.

Fig. 14 shows an image of the projector projected 20 degrees upward from the ground. Vertical keystoning occurs because the length of the projection distance of the upper part of the image increases. To compensate for this distortion, the image must be predistorted to reduce the size of the top side in the horizontal direction. However, if the image is calibrated only horizontally, it will appear to extend vertically, as shown in Fig. 15. To maintain the aspect ratio, it should be corrected not only horizontally but also vertically, and the results are shown in Fig. 16. At HD resolution, our distortion correction module can compensate for a vertical slope of up to 20 degrees considering the aspect ratio maintenance, and up to 47 degrees of vertical slope without considering the aspect ratio maintenance.

Fig. 17 shows the keystone in the horizontal direction, as it tilts 15 degrees to the left. This composite keystone distortion can be perfectly corrected as shown in Fig. 18.

The predistortion for the side-by-side 3D input format is also applicable considering the different optical distortion conditions of the left and right panels. Once the function for

FIGURE 16. Keystone correction in the horizontal and vertical directions in the distortion shown in Fig. 14.

FIGURE 17. Keystone distortion caused by tilting the projector 15 degrees to the left.

FIGURE 18. Keystone correction in the horizontal and vertical direction in the distortion of Fig. 17.

3D transformation is enabled, the half image for each side is individually horizontally interpolated to fit the panel resolution, and then the interpolated images are simultaneously transmitted to the panels. As the 3D contents are common for binocular NTE applications, this capability is useful.

TABLE 2. Total power consumption measured in display controller.

Resolution	Used LUT	Power [mW]		
		White	PM5644	Mountain
HD	Disable distortion correction module	186	188	195
	In Fig. 10	225	259	308
	In Fig. 11	235	266	316
	In Fig. 12	226	259	309
	In Fig. 16	218	253	302
	In Fig. 18	218	252	300
FHD	Disable distortion correction module	238	246	261
	In Fig. 10	304	357	424
	In Fig. 11	319	366	444
	In Fig. 12	305	357	424
	In Fig. 16	294	347	415
	In Fig. 18	294	346	412

TABLE 3. Test conditions for power measurement.

Conditions	Value
Input interface	Open LDI
Output interface	LVDS 4 channel for HD resolution LVDS 8 channel for FHD resolution
Input frame rate	60 Hz
Output field/frame rate	Around 360/120 Hz
Operating clock frequency	Around 125 MHz

Even though each optical system suffers from different distortions due to manufacturing issues, the proposed grid-based correction module can correct the distortions on the left and right sides, by applying different disparity information to the LUTs.

In addition to these distortions, if the LUT is set appropriately, the proposed logic can compensate for various distortions, such as decentering, rotating, or global or partial zooming.

The distortion correction module allows the user more comfortable viewing by providing a more natural image, but when this module is enabled, the power consumption of the display controller increases. TABLE 2 shows the total power consumption measured in display controller. The power is measured in HD and FHD resolution, various LUTs, and various images. The LUTs used for power consumption measurements are the same as those used to correct various distortions in the previous experiments. In the test images, 'White' represents a white monochromatic pattern, 'PM5644' is the image used in Fig. 9 to Fig. 12, and 'Mountain' is the image used in Fig. 14 to Fig. 18. The other conditions such as clock frequency and input/output interface are shown in TABLE 3.

Experimental results show that using the distortion correction module increases power consumption by from 31 to 109 mW at HD resolution and from 58 to 178 mW at FHD resolution. By percentage, the power consumption is increased by at least 17% and by up to 67%. In mobile products requiring low power consumption, this power consumption

may be considered high. However, if another chip based on GPU is used for the distortion correction function, it will consume more than 1W of power and the board size will increase. Therefore, for mobile application, it is more suitable to incorporate the distortion correction function in display controller than to use another chip that can process general graphics.

Also, the measurement results show that the power consumed by the distortion correction module is more dependent on the test image than on the type of LUT. The maximum difference in power depending on the test image is 125 mW, but the maximum difference in power according to the LUT is only 32 mW. This is because, if the LUT is complex, the computational complexity of the backward mapper will also increase. However, power consumption is more affected by changes in the data stored in the line buffers and block buffers.

VII. CONCLUSIONS AND FUTURE WORKS

Optical distortion and perspective distortion are common phenomena in microdisplay systems. In this paper, we proposed an architecture and an algorithm for real time correction of these complex distortions in microdisplay systems. By using the bilinear approximation with LUTs having disparity information, it is possible to correct the complex distortions simultaneously, and to reduce the complexity of the design, so that implementing the hardware does not become burdensome. We designed and implemented the correction logic on an ASIC, and then successfully demonstrated the logic with a microdisplay system using HD and FHD LCoS panels. The designed system can provide a compensated view to an end-user in real time with extremely low latency; hence, the user perceives a distortion-free image. The power consumption is also lower than other configurations, making it more suitable for wearable devices. This can enable the easy production of VR/AR contents, and less expensive NTE systems with desirable picture quality.

Even though the designed correction module is beneficial for various applications, some points also need to be improved, such as considering an automatic calibration process. In the current design, it may be difficult to find an appropriate LUT that perfectly compensates a complicated distorted view. In order to effectively use the proposed correction module, the assistance of a vision system may be needed for the process. Fortunately, many researchers have proposed auto-calibration systems using a camera. Evaluating the amount of distortion and calculating the proper LUT could be performed in an independent system.

Perspective distortion typically involves brightness distortion caused by the distance difference between the light source and the reflected point on a projection screen. If region-based brightness control is used, the overall picture quality can be improved.

The line memory of the distortion correction module occupies a large area. We may consider an architecture sharing frame memory without this line memory to reduce the size of the chip. To do this, the frame memory must support random access without performance degradation and support a higher data bandwidth for interpolation. In our display controller, it is more cost-effective to use additional line memory than to support these features, but it is worth to study designs that do not use additional line memory to miniaturize the chip.

REFERENCES

[1] J. Park, S.-C. Byun, and B.-U. Lee, "Lens distortion correction using ideal image coordinates," *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 987–991, Oct. 2009, doi: 10.1109/TCE.2009.5278053.

[2] Z. Zhang, Y. Matsushita, and Y. Ma, "Camera calibration with lens distortion from low-rank textures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Colorado Springs, CO, USA, Jun. 2011, pp. 2321–2328.

[3] C. A. Washburn, "A magnetic deflection up-date: Field equations, CRT geometry, the distortions and their corrections," *IEEE Trans. Consum. Electron.*, vol. 41, no. 4, pp. 963–978, Nov. 1995, doi: 10.1109/30.477213.

[4] T. Yoshinaga and K. Iwasaki, "A deflection yoke for projection TV with reduced geometrical distortion," *J. Soc. Inf. Display*, vol. 13, no. 8, pp. 619–625, Aug. 2005, doi: 10.1889/1.2039289.

[5] R. A. Petrozzo and S. W. Singer, "Cinema projection distortion," *SMPTE J.*, vol. 109, no. 8, pp. 648–651, Aug. 2000, doi: 10.5594/J05273.

[6] Y. Ye, "A new keystone correction algorithm based on edge-directed interpolation of the projector," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 8, no. 6, pp. 45–50 Jun. 2015, doi: 10.14257/ijsip.2015.8.6.06.

[7] M. Winzker and U. Rabeler, "Electronic distortion correction for multiple image layers," *J. Soc. Inf. Display*, vol. 11, no. 2, pp. 309–316, Jun. 2003, doi: 10.1889/1.1825661.

[8] S. W. Eo, J. G. Lee, W. Hong, Y. C. Ko, and M. S. Kim, "A novel architecture and algorithm for real-time correction of pincushion distortion in microdisplay systems," in *SID Int. Symp. Dig. Tech. Papers*, vol. 47. May 2016, pp. 1297–1300.

[9] D. Armitage, I. Underwood, and S. T. Wu, *Introduction to Microdisplays*. New York, NY, USA: Wiley, 2006, p. 1.

[10] D. Weng, Y. Wang, and Y. Liu, "GPU based real-time correction for optical distortions in head-mounted displays," in *Proc. 5th Int. Conf. Image Graph.*, Xi'an, China, 2009, pp. 672–676.

[11] OpenCV. *Camera Calibration and 3D Reconstruction*. Accessed: Apr. 3, 2018. [Online]. Available: https://docs.opencv.org/3.4.1/d9/d0c/group__calib3d.html

[12] ImageMagick. *ImageMagick v6 Example—Lens Correction*. Accessed: Apr. 3, 2018. [Online]. Available: http://www.imagemagick.org/Usage/lens/

[13] Gimp. *Chapter 4.8. Lens Distortion*. Accessed: Apr. 3, 2018. [Online]. Available: https://docs.gimp.org/en/plug-in-lens-distortion.html

[14] K. V. Asari, "Design of an efficient VLSI architecture for non-linear spatial warping of wide-angle camera images," *J. Syst. Archit.*, vol. 50, no. 12, pp. 743–755, Dec. 2004, doi: 10.1016/j.sysarc.2004.05.001.

[15] H. T. Ngo and V. K. Asari, "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 436–444, Mar. 2005, doi: 10.1109/TCSVT.2004.842609.

[16] P. Y. Chen, C. C. Huang, Y. H. Shiau, and Y. T. Chen, "A VLSI implementation of barrel distortion correction for wide-angle camera images," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 1, pp. 51–55, Jan. 2009, doi: 10.1109/TCSII.2008.2010165.

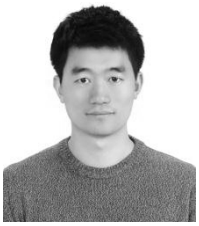
[17] S.-L. Chen, H.-Y. Huang, and C.-H. Luo, "Time multiplexed VLSI architecture for real-time barrel distortion correction in video-endoscopic images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1612–1621, Nov. 2011, doi: 10.1109/TCSVT.2011.2129850.

[18] H.-S. Jeong and T.-H. Kim, "An efficient processor for joint barrel distortion correction and color demosaicking," in *Proc. IEEE Int. Symp. Circuits Syst.*, Lisbon, Portugal, May 2015, pp. 1782–1785.

[19] C. Ricolfe-Viala and A.-J. Sanchez-Salmeron, "Lens distortion models evaluation," *Appl. Opt.*, vol. 49, no. 30, pp. 5914–5928, Oct. 2010, doi: 10.1364/AO.49.005914.

[20] D. Eadie, F. Shevlin, and A. Nisbet, "Correction of geometric image distortion using FPGAs," *Proc. SPIE*, vol. 4877, pp. 28–38, Mar. 2003. [Online]. Available: https://spie.org/Publications/Proceedings/Paper/10.1117/12.463765

[21] K. T. Gribbon, C. T. Johnston, and D. G. Bailey, "A real-time FPGA implementation of a barrel distortion correction algorithm with bilinear interpolation," in *Proc. Image Vis. Comput., Palmerston North, New Zealand*, Nov. 2003, pp. 408–413.
[22] J. R. Ryoo, E. S. Lee, and H. K. Park, "Real-time implementation of an LUT-based image warping system," in *Proc. Int. Symp. Robot.*, Seoul, South Korea, Oct. 2013, pp. 1–4.
[23] L. Qiang and N. M. Allinson, "FPGA-based optical distortion correction for imaging systems," in *Proc. 8th Int. Conf. Signal Process.*, Beijing, China, Nov. 2006, pp. 16–20.
[24] H. Blasinski, W. Hai, and F. Lohier, "FPGA architecture for real-time barrel distortion correction of colour images," in *Proc. IEEE Int. Conf. Multimedia Expo*, Barcelona, Spain, Jul. 2011, pp. 1–6.
[25] H. T. Ngo and V. K. Asari, "Developing a FPGA-based high performance, power-aware architecture for the correction of radial lens distortion in video stream," *Int. J. Program. Devices Circuits Syst.*, vol. 7, no. 1, pp. 33–41, May 2007.
[26] Z. Jereb and J. Diaci, "Real-time keystone correction of video image using FPGA," *Informacije MIDEM*, vol. 40, no. 3, pp. 204–207, Sep. 2010.
[27] F. Blaicher, J. Yang, and R. Stiefelhagen, "Automatic distortion correction for a full windshield head-up display system," Faculty Comput. Sci., Karlsruhe Univ., Karlsruhe, Germany, Tech. Rep., Jan. 2009. [Online]. Available: http://isl.anthropomatik.kit.edu/cmu-kit/downloads/StA_Blaicher.pdf
[28] K.-T. Teng *et al.*, "Reduction of possible flicker and color breakup using deflicker-FSC method for field-sequential-color displays," in *SID int. Symp. Dig. Tech. Papers*, vol. 47. May 2016, pp. 809–812.
[29] *SMIA 1.0 Part 5: Camera Characterisation Specification, Version 1.0*, Nokia and ST, Espoo, Finland, Jun. 2004, p. 61.
[30] *Basler Ace User's Manual for USB 3.0 Cameras, Version 10*, Basler AG, Ahrensburg, Germany, Jan. 2017.
[31] Konmann GmbH. *KMN-FA1601C Product Description*. Accessed: Apr. 3, 2018. [Online]. Available: http://konmann.com/index.php/Product-view-id-142.html



SUNG-WOOK EO received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2010. He is currently pursuing the Ph.D. degree with the School of Electrical Engineering, Korea University.

Since 2013, he has been with the Research and Development Group, RAONTECH, where he is currently a Research Engineer. His current research interests include signal interface, image processing, and low power architecture in the microdisplay systems.

JOON GOO LEE received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2005, 2008, and 2015, respectively. He actively researched passive RFID systems and their new applications, ultra low-power wireless communication systems, and efficient protocols at Korea University with global companies such as Samsung Electronics and LG Electronics. Since 2015, he has been a Senior Engineer with the Research and Development Group, RAONTECH, where he is responsible for the digital design of microdisplay systems. His current research interests include high definition microdisplay panel design, low-power timing controller architecture of the panel, and data compression and high speed interface for video streaming.

MIN-SEOK KIM received the B.S. and M.S. degrees from Korea University in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering from Dankook University in 2011.

He was the Key Engineer with Samsung SDI, Iljin Display, developing display driving circuits and HTPS TFT LCD, and LCoS display module for microdisplay applications. Since 2012, he has been the Head of the Research and Development Group, RAONTECH. He oversees the engineering departments and focuses on developing new products and technologies. He has 25 years of industry experience in micro display circuit design.

YOUNG-CHAI KO received the B.Sc. degree in electrical and telecommunication engineering from Hanyang University, Seoul, South Korea, and the M.S.E.E. and Ph.D. degrees in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1999 and 2001, respectively. He was a Research Scientist with Novatel Wireless in 2001. In 2001, he joined Texas Instruments, Inc., Wireless Center, San Diego, CA, USA, as a Senior Engineer. He is currently a Professor with the School of Electrical Engineering, Korea University. His current research interests include the design and evaluations of multi-user cellular system, MODEM architecture, mm-wave, and terahertz wireless systems.