A Measurement-based Active Distortion Compensation for 3D Head-Up-Display

Akinori Sato, **Ayuki Hayashishita, Kaoru Kusafuka**

akinori.satou.cy@kyocera.jp
Kyocera Corporation, Shiga, Japan
Keywords: Head-Up-Display, 3D HUD, Polynomial approximation, Distortion compensation

ABSTRACT

Since automotive HUD has a display distortion changing depending on eye position, a 2D HUD has a problem of binocular disparity caused by the difference in distortion between the right and left eye positions. Using a parallax barrier 3D HUD with eye tracking, we show that the above problem can be solved by performing proper 3D image composition and distortion compensation according to each of the left and right eye positions as 3D HUDs active distortion compensation.

1 Introduction

Head-up-Display (HUD) has the excellent feature of displaying a virtual image display several meters in front of the driver, minimizing eye movement and focus adjustment while driving.

On the other hand, HUD optical system includes a distorted windshield, which causes a large distortion of the shape of the virtual image display. This distortion and displacement of the virtual display depends on the position of the eye that is viewing the HUD. So-called active distortion compensation, which corrects distortion at eye position, is an effective countermeasure against this eye position-dependent distortion. However, due to the difference in position between the right and left eye, there remains a misalignment between the virtual image seen by the right eye and one of the left eye. This is unintended binocular disparity due to distortion and known as one of the display quality problems of HUD [1]. A parallax barrier 3D HUD with eye tracking [2, 3] may have a possibility to address the problem of unintended binocular disparity by individually correcting the distortion in the images shown to the left and right eye.

A technique to perform appropriate distortion compensation for arbitrary eye positions in a HUD has been reported by Folker [4]. In this technique, the HUD display is captured at different camera (eye) positions, and the amount of distortion at several locations on the LCD coordinates is extracted for each eye position. And polynomial is used to approximate the distortion and to do distortion compensation.

The possibility of applying this technique to a 3D HUD is mentioned [4], and we actually have applied this distortion compensation to a parallax barrier 3D HUD in this paper.

For the part of the 3D image composition process in the 3D HUD, we had developed a technique to measure the information necessary for image composition by changing the camera (eye) position as well as distortion compensation, and to implement it in a polynomial equation [5].

3D image composition using polynomials and distortion compensation using polynomials are compatible and can be described in a unified manner. We confirmed that distortion compensation and 3D image composition according to eye position can be performed on a 3D HUD to display appropriate 3D images and eliminate unintended binocular disparity caused by distortion in real time.

2 Method

2.1 Problem statement

In a 2D HUD, binocular disparity caused by the difference in distortion between the right and left eye positions is a problem. We will report on a technique to eliminate the binocular disparity by combining 3D images and applying distortion compensation for each left and right eye position in a parallax barrier 3D HUD with eye tracking.

2.2 3D image composition

For any given eye position, it is necessary to determine which LCD pixels are visible to the right eye or to left eye. A measurement-based technique to do this has been developed by us [5]. The information (phase) required for 3D image composition is derived by taking pictures of the HUD display showing predetermined patterns at multiple camera (eye) positions and analyzing the images. The phase is expressed as a polynomial $P(E_x, E_y, E_z, L_x, L_y)$ with the right eye positions(E_x, E_y, E_z) and LCD pixel coordinates (L_x, L_y) as variables.

This polynomial allows the phase to be reproduced at any eye position, and the pixels visible from the respective positions of the right and left eyes can be isolated in sub-pixel units. The phase represented by the polynomial can be described by shader language in a unified manner with distortion compensation, which is also represented by a polynomial, as described below. In this paper, we describe the distortion compensation.

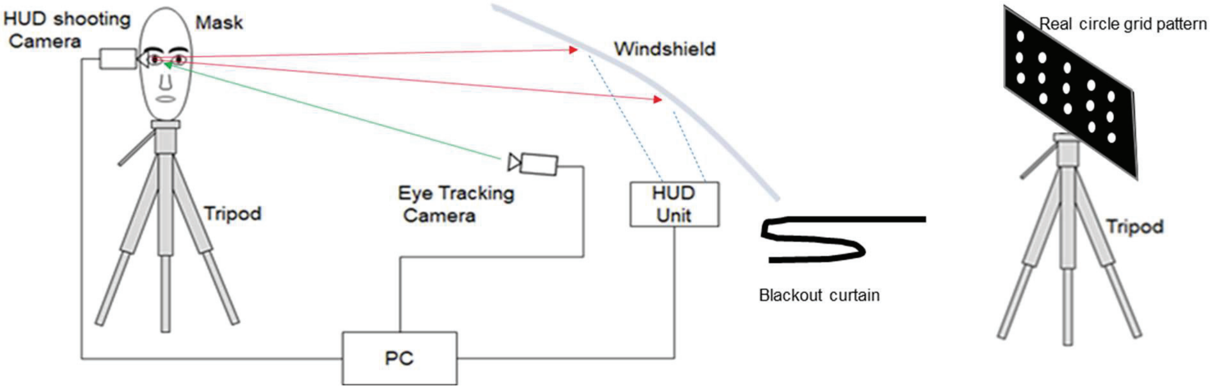

Fig.1 Distortion measurement system

2.3 Virtual display distortion measurement system

Figure 1 shows the measurement system. 3D HUD apparatus consists of an eye tracking camera, a HUD unit, and a windshield. The HUD unit includes an LCD with a parallax barrier and a magnifying mirror. The eye tracking camera acquires the position coordinates of the HUD shooting RGB camera. The LCD displays a predetermined circle grid pattern. Real circle grid with a circle grid printed in the size of the HUD virtual display is placed at the position of the HUD virtual display. The mask is needed for the eye tracking camera to detect the eye position. The RGB camera is fixed to the right eye position of the mask and shoots the HUD virtual display and the real circle grid. PC controls the measurement imaging and HUD display.

2.4 Circle grid

A circle grid of 5x7 circles is displayed on the LCD (Figure 2). The real thing circle grid pattern, printed in the same size as the virtual display, is placed at the virtual display position (2.5 m from the HUD device in this case).

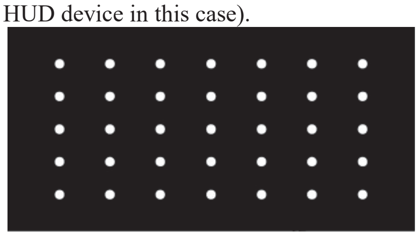
Fig. 2 Circle grid image displayed on LCD

The coordinates of each circle position on the image from the captured images of the virtual display and the real thing circle grid pattern can be easily obtained with OpenCV, an open-source computer vision library.

2.5 Shooting procedure

After fixing the RGB camera position, the eye tracking camera get the RGB camera position (E_x, E_y, E_z) as right eye of the mask. Then, with the camera position fixed, the virtual display of the circle grid and the real object are captured; when shooting the HUD display, a blackout curtain is draped over the windshield so that the real object cannot be seen, and when shooting the real circle grid, the HUD display is turned off.

The above process from fixing the RGB camera position to shooting is performed by changing the camera position at, for example, 27 different locations. A precise XYZ stage is not necessary for moving the RGB camera, and the camera may be fixed to a simple tripod and moved as shown in the figure. It is important that the camera position is fixed not moved and the image is captured; camera orientation and lens distortion do not affect the estimate of the display displacement in the next step.

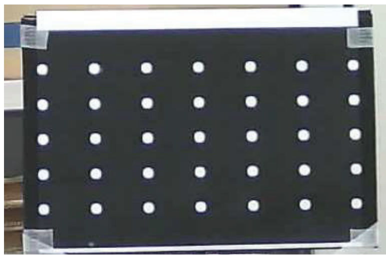
Fig. 3 Example of captured real thing

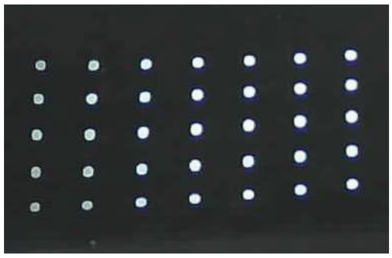
Fig. 4 Example of captured virtual display

2.6 Estimating displacement from captured images

Since display distortion can be expressed as a distribution of displacements in the x- and y-directions, we measure the displacements in the x- and y-directions of circles.

The displacement can be estimated by the difference between the position of each circle on the circle grid of the real thing and virtual displays taken at each eye position. There are three types of circle grids that appear here: those displayed on the LCD, those taken of the real object, and those taken of the virtual display. The meaning and notation of the position coordinates of these circles will be explained. The coordinates of circles displayed on LCD are $(LC_x, LC_y)_i, i = 1, \ldots 35$where i represents 35 circles. The resolution of the LCD used here is 1280x640xRGB, so the center coordinates of each of these circles are in the range$(0,0) - (1280,640)$. The coordinates of the circles on captured image of the real thing circle grid (Figure 3) are $(GR_x, GR_y)_{i,j}$ where j represents camera positions. Let $(GV_x, GV_y)_{i,j}$ denote the coordinates of the circles on capture image (Fig. 4) of the circle grid on the virtual display.

Since the positions of the RGB cameras are fixed, the difference in the positions of the real and virtual display circle grids on the captured image reflects the display distortion. This difference in position can be easily obtained as pixel coordinates on the captured image using OpenCV as described above. By converting this circle displacement to LCD coordinates, we can estimate the distortion of the virtual display in LCD coordinates. In other words, for a pixel at position$(LC_x, LC_y)_i$, we know how many pixels we need to correct on the LCD for distortion compensation.

This can be accomplished by finding a polynomial that converts the pixel coordinates on the captured image to LCD coordinates. From the set of known position coordinates $(LC_x, LC_y)_i$ of the circle on the LCD and position coordinates $(GV_x, GV_y)_{i,j}$, $i = 1, \dots 35$ of the circle extracted from the virtual captured image, the functions $Lx_i = F\_Lx(GVx, GVy)_i$ and $Ly_i = F_Ly(GVx, GVy)_i$in polynomial are determined using the least square method. Here we used a third-order polynomial, which consists of 10 terms, $\mathrm{F(x,y)} \approx a_1 + a_2\mathrm{x} + a_3\mathrm{y} + a_4x^2 + a_5\mathrm{xy} + a_6y^2 + a_7x^3 + a_8x^2y + a_9yx^2 + a_{10}y^3$, written using symbols as follows. Let $(\mathrm{x,y})$ be the coordinates (GV_x, GV_y)on the captured image.

$$F(x, y) \approx \sum_{\substack{d_1, d_2 \\ d_i \in \mathbb{N}_0 \\ \sum d_i \leq 3}} a_{k(d_1, d_2)} x^{d_1} y^{d_2}$$

Using the least-squares method, determine 10 coefficients so that $\sum(LCx_i - F\_Lx(GVx, GVy)_i)^2$ is minimized. This function can be used to convert any position on the captured image to coordinates on the LCD. The amount of displacement of each circle in the LCD coordinates is obtained as $LCx_i - F\_Lx(GRx, GRy)_i$ in the x-direction and $LCy_i - F\_Ly(GRx, GRy)_i$ in the y-direction. This amount of display displacement in LCD coordinate is denoted as $(\mathrm{Ldx, Ldy})_i$.

2.7 Active distortion compensation

Consider finding a polynomial that reproduces the amount of displacement at any given eye position. In other words, from set of eye position coordinates $(\mathrm{Ex, Ey, Ez})_j$, circle position $(\mathrm{LCx, LCy})_{i,j}$ and displacement amount data $(\mathrm{Ldx, Ldy})_{i,j}$, we find a polynomial that represent displacement. Here we use a polynomial of second order with respect to $(\mathrm{Ex, Ey, Ez})_j$ and third order with respect to$(\mathrm{Lx, Ly})_i$. This polynomial consists of 46 terms and is written as follows where $(\mathrm{u, v, w})$ are the eye position coordinates $(\mathrm{Ex, Ey, Ez})$ and $(\mathrm{x, y})$ are the LCD coordinates$(\mathrm{Lx, Ly})$.

$$F(u, v, w, x, y) \approx \sum_{\substack{d_1, d_2, d_3, d_4, d_5 \\ d_i \in \mathbb{N}_0 \\ d_1, d_2, d_3 \leq 2 \\ \sum d_i \leq 3}} a_{k(d_1, d_2, d_3, d_4, d_5)} u^{d_1} v^{d_2} w^{d_3} x^{d_4} y^{d_5}$$

The polynomial, $\mathrm{Ld}x_{i,j} = F\_Ldx(Ex, Ey, Ez, LCx, LCy)_{i,j}$ and $\mathrm{Ld}y_{i,j} = F\_Ldy(Ex, Ey, Ez, LCx, LCy)_{i,j}$ are determined by the least square method. The reason for using second order with respect to eye position here the distortion is varying relatively simply with respect to eye position and to reduce the number of camera positions to avoid overfitting. In the case of second order eye position, three camera positions in the X, Y, and Z directions (3x3x3=27) are required to avoid overfitting, and third order in LCD coordinates are used to compensate barrel distortion.

2.8 Implementation

Distortion compensation and 3D image composition processing in 3D HUD operation are performed following eye position detection (Figure 5). Substituting the eye position into the polynomials $\mathrm{Ldx(Ex, Ey, Ez, Lx, Ly)}$ and $\mathrm{Ldy(Ex, Ey, Ez, Lx, Ly)}$ yields the reduced polynomials $\mathrm{P(Lx, Ly)}$, $\mathrm{LDx(Lx, Ly)}$ and $\mathrm{LDy(Lx, Ly)}$. The coefficients of each polynomial are passed to the GPU for per-sub-pixel processing in the fragment shader. The GPU performs the processing for each pixel, while the CPU is designed for very light processing, just preparing the data to be processed by the GPU.

The shader first determines whether to use the picture for the left eye or the right eye based on the phase value of the coordinates of the sub-pixel to be rendered. Once the picture to be used is determined, the amount of displacement is calculated and the color of the position shifted by the amount of displacement from the texture data of the content image is adopted to achieve appropriate distortion compensation. By the above operations, it is possible to display separate distortion-compensated images for the left and right eye separately.

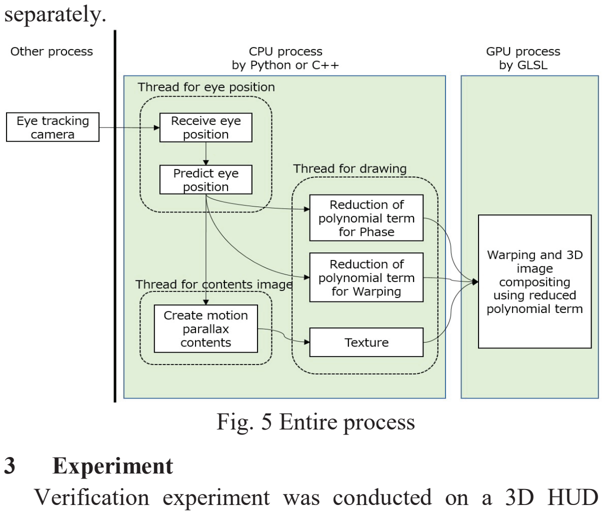

Fig. 5 Entire process

3 Experiment

Verification experiment was conducted on a 3D HUD device, 1280x640xRGB, 4" LCD. Distance to virtual display is 2.5 m, FoV 10 degrees. The windshield was used from a production car, Lexus: GS350; GS350 uses a 1.8-inch 2D

HUD, which was diverted to the 3D HUD experiment this time. A camera was attached to the right eye position of the mask to take pictures of the HUD display and real circle grid. And the camera position was moved three places in the X, Y, and Z directions, centering on the optimum viewing position, for a total of 27 locations. The range of camera movement in each direction was 140 mm in range in the X direction, 60 mm in the Y direction, and 240 mm in range centered at 530 mm in the Z direction. The eye position coordinates are X: horizontal, Y: vertical, and Z: depth, based on the eye tracking camera position. Active distortion compensation of the 3D HUD was operated on this device by performing steps 2.5-2.8 above. The processing required for the display was not much different from that of a normal parallax barrier 3D display, and it was confirmed that the in-vehicle SoC, R-Car H3, could perform real-time processing at 60 fps.

3.1 Confirmation of distortion compensation

Figure 6 shows the captured image of the virtual display of the circle grid after distortion compensation at the camera position where Figures 3 and 4 were taken. It is clear that the distortion compensation has been successfully performed, as the images match well with the real thing circle grid taken. It was confirmed that the position of the circle grid at the other camera positions also agreed well with that of the real object.

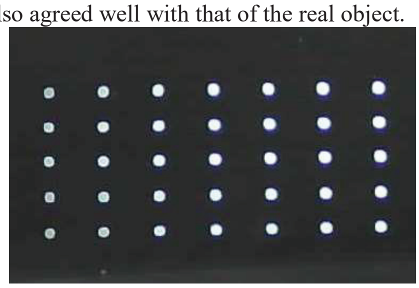

Fig. 6 Captured image of virtual display after distortion compensation (camera position same as Figure 3, 4)

3.2 Confirmation of binocular disparity due to distortion

In the HUD used in the experiment, we saw the displacement of the circle at the lower left edge was large when the eye position was changed. The displacement of the circle at the lower left edge was evaluated using the method described in 2.6 when the camera (eye) position was set to -30 mm in the Y direction and 500 mm in the Z direction and moved in the X direction. The amount of displacement in the X-direction and the amount of displacement in the Y-direction before and after distortion compensation were plotted using LCD pixels as units (Figure 7). Before correction, the position of the circle changed according to the eye position. Since the left eye is located at the position where the right eye position is shifted by -65 mm in the X direction. The difference in the amount of displacement when viewed with a horizontal axis 65 mm wide in this graph is unintended binocular disparity. After compensation, the circle displacement due to eye position is suppressed, indicating that the binocular disparity problem is eliminated.

In the experiment, a misalignment of ±3 pixels remains after compensation. This may be due to the effect of the accuracy of a series of image processing or the fact that it was not fully approximated by the polynomial equation used here.

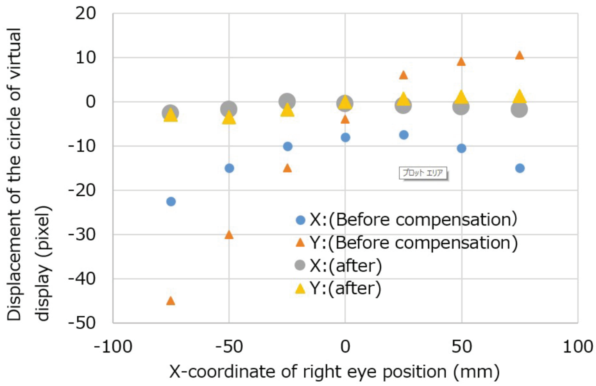

Fig.7 Displacement (pixel) of the lower left edge circle to the X-coordinate (mm) of eye position.

4 Conclusions

We described a method to do appropriate distortion compensation to view of right-eye and left-eye using a parallax barrier 3D HUD with eye tracking. Display displacement was reproduced in polynomial form and incorporated into the 3D HUD system, and it was confirmed that active distortion compensation in the 3D HUD can suppress unintended binocular disparity. This method is expected to be optimal as a base for AR-HUD. Although we reported on the parallax barrier system, the same method can be applied to the lenticular 3D HUD.

References

[1] D. Wagner et al., "Impact Study of Windshield Geometry on the Subjective Customer Perception for Augmented Reality Head-up Displays (AR HUD)", SID'20, 254-257, (2020).

[2] D. Suzuki et al., "A Wide View Glass-less 3Display with Head-Tracking System for Horizontal and Vertical Directions", SID'16, 990-993, (2016).

[3] H. Nakamura et al., "A Novel Eye Tracking System to Expand Viewing Area in all Directions for Glasses-Free 3D Display Displayable in both Portrait and Landscape Modes," Proc. IDW '18, 792-795 (2018).

[4] Folker Wientapper et al., "A Camera-Based Calibration for Automotive Augmented Reality Head-Up-Displays." In IEEE International Symposium on Mixed and Augmented Reality 2013 Science and Technology Proceedings, 189-197, 2013.

[5] A. Sato et al., "A Measurement-Based Image Compositing for 3D Head-Up-Display", SID'22, 708-711, (2022).