

# *Real-Time Computational Visual Aberration Correcting Display Through High-Contrast Inverse Blurring.*

Akhilesh Balaji<sup>1</sup>, \* and Dhruv Ramu<sup>1</sup>, †

<sup>1</sup>*Neev Academy, Bengaluru, Karnataka, 560037 India*

(Dated: October 12, 2024)

This paper presents a framework for developing a live vision-correcting display (VCD) to address refractive visual aberrations without the need for traditional vision correction devices like glasses or contact lenses, particularly in scenarios where wearing them may be inconvenient. We achieve this correction through deconvolution of the displayed image using a point spread function (PSF) associated with the viewer's eye. We address the issue of ringing artefacts by introducing a masking technique applied to the prefiltered image. We also enhance the display's contrast and reduce color distortion by operating in the YUV/YCbCr color space, where deconvolution is performed solely on the luma (brightness) channel. Finally, we introduce a novel technique to calculate a real-time PSF that adapts based on the viewer's spherical coordinates relative to the screen. This ensures that the PSF remains accurate and undistorted even when the viewer observes the display from non-perpendicular angles, thereby providing consistent visual correction regardless of the angle of view. The results of our display demonstrate significant improvements in visual clarity, achieving a structural similarity index (SSIM) of 83.04%, highlighting the effectiveness of our approach.

---

\* [akhilesh.balaji.bangalore@gmail.com](mailto:akhilesh.balaji.bangalore@gmail.com)

† [dhruvramu@gmail.com](mailto:dhruvramu@gmail.com)

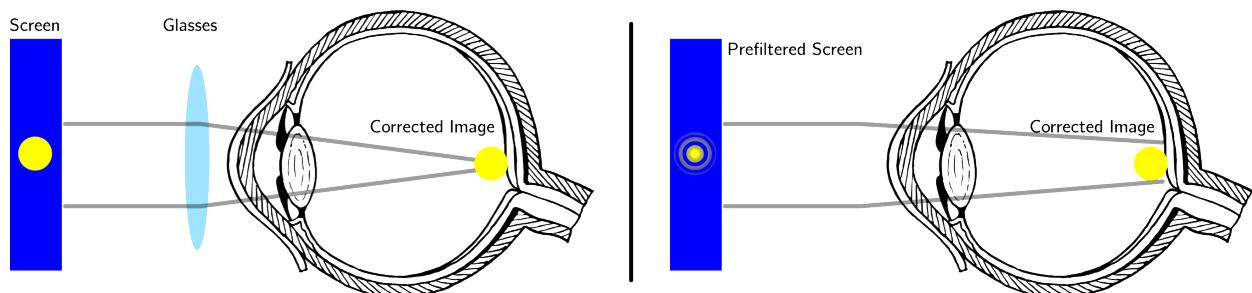
## I. INTRODUCTION

Today, 2.2 billion people have near or distance visual aberrations across all ages. Visual impairment is not just a health problem; annual global costs associated with visual impairment based treatment exceed 411 billion USD [1]. Myopia is a type visual impairment where objects at a relatively close distance are not visible (nearsightedness) due to the convergence of light rays after the retina. It is expected to affect 50% of the world population by 2050. In hyperopia, light rays converge before the retina, resulting in farsightedness [2].

Traditional solutions to defocus are spectacles and contact lenses; they do not provide perfect solutions to the widespread visual aberration problem. Spectacles are incompatible with virtual and augmented reality platforms as they cannot be worn in conjunction with a VR headset or 3D anaglyph glasses. Glasses may not be necessary when adventure sports are being conducted, instead simply getting in the way, especially if the patient does not have use for the devices outside reading or working. Blurry or dusty eyeglasses may cause further eye strain, aggravating the visual aberration they were diagnosed with. The use of contact lenses to solve this may lead to the progressive dryness of eyes [3].

### Problem Statement

Glasses and contact lenses provide solutions to lens-based visual aberrations, but for higher-order aberrations, lens manufacturing becomes difficult, or heavy/asymmetrical lenses may have to be dealt with. Thus, we ‘shift’ the glasses computationally onto the screen, eliminating the need for lenses to correct the aberration.



## II. BACKGROUND

The problem of vision correcting displays has primarily been addressed, in recent times, by the creation of light field displays. A light field captures the exact configuration of electromagnetic waves, complete with phase and amplitude descriptions, in a given optical system. In Huang’s 2014 paper [4], we see that the light field surrounding a display as perceived by a visually aberrated individual can be computed by passing the light field through computational lenses to simulate the aberration and its correction. The physical generation of this light field is achieved through a *light field display*—Huang et al. use pinholes placed over a pre-distorted image, such that the diffracted individual light fields emerging from each pinhole interfere with each other to produce the correct light field. This naturally implies the existence of a hardware component to achieve the light field propagation, which would have significant cost associated with it if it were to be deployed on a large-scale basis. Likewise, the authors state that the pre-processing procedure is computationally expensive and requires high processing time.

Alternatively, one may use a holographic approach [5]. In a traditional hologram, the light field is captured on a photosensitive film by means of a light source passing over the objects to be recorded that interferes with a second coherent light beam used to capture phase difference. When exposed to the original coherent light beam once more, the light field is regenerated around the film and the three-dimensional scene is captured [6]. However, upon placing an appropriate lens simulating myopia or hyperopia over the scene, the blurred (or de-blurred, in the case of a visually aberrated viewer) nature of the light field is also captured. This allows for vision correction. However, this only works for still images, and techniques for temporally evolving holograms are still under research [7].

In our work, we adopt the technique of Alonso and Barreto [8], wherein no hardware component is required to deblur an image. The techniques we use for optimization and colour restoration may be extended to any kind of deconvolution algorithm. While the method proposed by Alonso and Barreto is also computationally expensive, we propose methods of optimizing these algorithms to create a *real-time* vision correcting display,

which the user can interact with.

### III. METHOD

#### A. General Overview

As can be seen in Figure 1, we take snapshots of the on-screen image at small intervals of time, deconvolve these according to the position of the viewer estimated at that point in time, and display this on the screen, replacing the original image.

#### B. Vision Correction Techniques

We use a fundamental operation in image processing, that is convolution, to simulate the blurring of the eye. The eye is characterized by a blur kernel that is represented in the point spread function (PSF; or optical transfer function, OTF) associated with it. We define the PSF to be:

$$k(x, y) := \begin{cases} \chi & \text{if } \sqrt{x^2 + y^2} \leq r, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Here,  $\chi$  is chosen such that:

$$\iint_{-\infty}^{+\infty} k(x, y) \, dx \, dy = V_k = 1;$$

that is, the PSF integrates to unity such that no light is lost, nor is it gained. The image brightness is conserved. This is called normalization. This integral becomes a discrete summation when dealing with images—discrete functions may thus be represented as matrices or 2D arrays. The PSF model, as obviated above, is that of a disk, which best models defocus in the human eye, due to spurious resolution induction, as will shortly be seen.

The PSF may also be defined using Zernike polynomial aberration representation, if the polynomial parameters are empirically determined. The aberration function is defined as the sum of the Zernike polynomial individual aberrations that compose it, as  $\mathcal{Z}(x_\alpha, y_\alpha) = \sum_j Z^j(x_\alpha, y_\alpha)$ , where  $Z^j \in \mathcal{Z}$  are Zernike polynomials;  $(x_\alpha, y_\alpha)$  are coordinates on the pupil aperture plane. Fourier optics tells us that the PSF must thus be

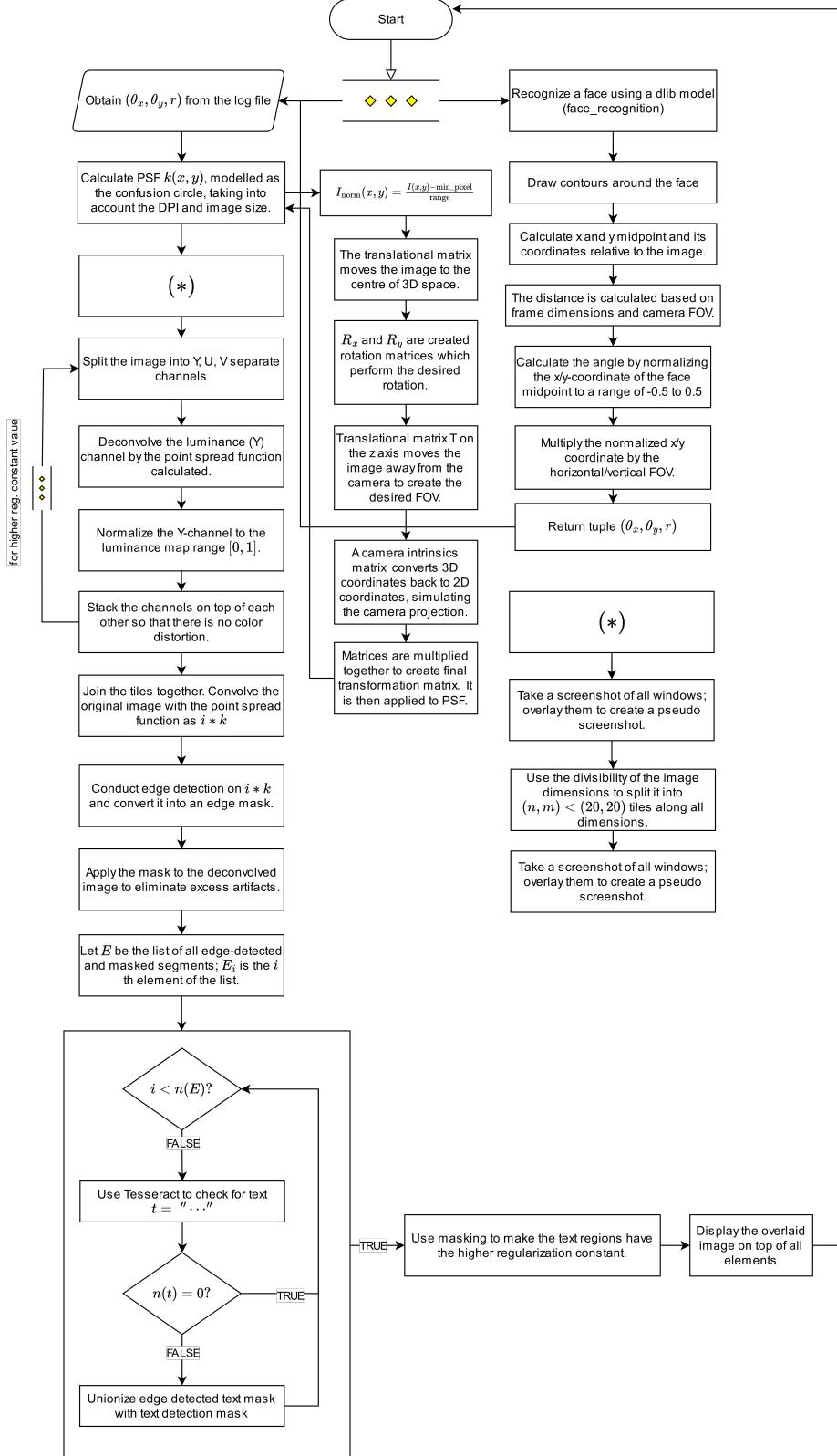


FIG. 1. Flowchart detailing the basic design.

given by the Fourier transform:

$$k(x, y) := \left| \iint_{-\infty}^{+\infty} A(x_\alpha, y_\alpha) \cdot e^{-i2\lambda^{-1}\pi\mathfrak{Z}} \cdot e^{-i(xx_\alpha + yy_\alpha)} dx dy \right|^2, \quad (2)$$

where  $A(x_\alpha, y_\alpha)$  is a pupil amplitude function that is unity within the pupil in the pupil aperture plan, and null everywhere else.

One then defines the blurring of the original image  $I(x, y)$  to be:

$$\hat{B}(\omega x, \omega y) := \hat{I}(\omega x, \omega y) \cdot \hat{k}(\omega x, \omega y) \quad (3)$$

$$\implies B(x, y) = \mathcal{F}^{-1} \{ \hat{I}(\omega x, \omega y) \cdot \hat{k}(\omega x, \omega y) \}. \quad (4)$$

Here,  $\omega$  is the angular frequency of the fourier tranforms, and  $\mathcal{F}$  is the fourier transform operation.

From here, we may define a function  $h(x, y)$  that inverts the operation above, such that:

$$\mathcal{F}^{-1} \{ \hat{I}(\omega x, \omega y) \cdot \hat{k}(\omega x, \omega y) \cdot \hat{h}(\omega x, \omega y) \} = I(x, y).$$

That is,  $h(x, y)$  and  $k(x, y)$  cancel each other out to give the *perfect PSF*, that is a bright single pixel in the origin. We define  $h(x, y) * k(x, y) = \mathbb{1}(x, y)$ , where  $\mathbb{1}$  denotes the identity. Note that:

$$k(x, y) * h(x, y) \equiv \mathcal{F}^{-1} \{ \hat{k}(\omega x, \omega y) \cdot \hat{h}(\omega x, \omega y) \},$$

and  $*$  is the convolution operator.

The recovery of  $I(x, y)$  from  $B(x, y)$  is the process of *deconvolution*. This is not as simple as it seems, for in defining  $h(x, y)$  as follows:

$$\hat{h}(\omega x, \omega y) = \begin{cases} \hat{k}(\omega x, \omega y)^{-1} & \text{if } x \geq \epsilon, \\ 0 & \text{otherwise,} \end{cases}$$

We introduce severe artefacts in the convolution (that is a deconvolution by  $k(x, y)$ ), and in the introduction of negative pixel intensities.

Here we note that by deconvolving an on-screen image,  $I(x, y)$ , by the PSF of the human eye,  $k(x, y)$  will, theoretically, define a pre-distorted image  $P(x, y)$ , such that the

defocused eye perceives it clearly through the convolution  $P(x, y) * k(x, y) = I(x, y)$ , that is the original image.

Recall from (1) that  $r$  was the variable upon which the PSF is dependent. We take  $r$  to be:

$$r = a \frac{|d_f - d_0|}{d_f}.$$

Here,  $a$  is the pupil diameter;  $f$  is the focal length;  $d_o$  is the distance from the display;  $d_f$  is the distance to the plane of focus.  $d_f$  can be computed using the thin lens equation:

$$\frac{1}{d_f} + \frac{1}{d_e} = \frac{1}{f},$$

where  $d_e$  is the eye's internal diameter.

In the created prototype, we have made use of Wiener deconvolution to minimize noise.

As the user is susceptible to movement while viewing the screen, they will continue to view the screen with PSF  $k(x, y)$ , but the on-screen PSF must be perspective transformed to ensure that spatial and angular ratios are conserved from all angles—in the case of the circle of confusion, it must appear circular from all angles, for the deconvolution to be effective. We use the coordinates, and dimensions of the eye to create a 2D transformation matrix encoding the perspective transformation to the PSF's current perception. We then apply the left inverse matrix to the original PSF, obtaining the perspective-corrected image.

This is then integrated with live distance and angle tracking, so that at discrete intervals, the PSF will be updated to reflect changes in distance and angle from the screen.

### C. Live GUI Layer

We take in the user's optical prescription in the beginning, assuming simple defocus, so that we can approximate the PSF using (1). We then create a layer on top of the operating system that continuously takes screen captures of what is behind the layer, and feeds it as an input to the deconvolution algorithm at discrete intervals. The output of the deconvolution algorithm is then used as a buffer for what is displayed

on the VCD layer. We use double buffering to reduce lag. Thus, the user is able to interact with a vision-corrected operating system.

#### D. Live Video Deconvolution

We have adapted our algorithm to work for video inputs as well. The prototype player allows interactive navigation through a video, including play/pause functionality, seeking, and caching frames for smoother playback. The player utilizes multi-threading for parallelization, with one thread handling caching, and another for rendering the frames and managing user interactions. When a video is inputted, upcoming frames (approximately 3 seconds' worth) are deconvolved and cached. This allows for efficient playback of videos. On an i9 32 GB RAM computer, it took 20 seconds to deconvolve 30 seconds of video or 1800 frames (90 FPS), whilst 27 FPS was observed on an i7 16GB RAM computer.

#### E. Ringing Artefact Reduction

We notice multiple ringing artefacts in a traditionally deconvolved image. A technique to remove majority of them that are not in the focus of the image was previously established by Alonso and Barreto, 2003, as follows.

First, the blurred on-screen image  $I(x, y) * k(x, y)$  is edge detected. An edge detection mask with a black fill,  $\mathbf{M}(x, y)$  is produced. The mask is then applied to the deconvolved image  $I(x, y) * h(x, y)$ , producing  $\mathbf{M}(x, y) \otimes (I(x, y) * h(x, y))$ .  $\otimes$  denotes logical intersection, and  $\oplus$  denotes logical conjunction. The inverse mask  $\mathbf{M}^{-1}(x, y)$  is applied to the original image as  $\mathbf{M}^{-1}(x, y) \otimes I(x, y)$ , and the two images are superimposed as:

$$(\mathbf{M}^{-1}(x, y) \otimes I(x, y)) \oplus [\mathbf{M}(x, y) \otimes (I(x, y) * h(x, y))].$$

This is the image then displayed on-screen.

We introduce an intermediate step, where each edge-detected segment is analysed for text presence. If text is present, we deconvolve that segment separately, with a higher

regularization constant, as it has more edges. We obtain, thus:

$$[\mathbf{M}^{-1}(x, y) \otimes I(x, y)] \oplus \left( \bigoplus_{\partial I} \begin{cases} \partial I * h(x, y, \rho_1) & \text{if } t(\partial I) = \emptyset, \\ \partial I * h(x, y, \rho_2) & \text{otherwise.} \end{cases} \right),$$

as the on-screen image, where  $\rho_2 > \rho_1$ , and  $t(I)$  returns the text present in a given image ( $\emptyset \equiv \text{" "}$ ). In this case, the function is supplied by Tesseract OCR models.  $\partial I$  are the individual edge detected and masked components of the image, such that:

$$\bigoplus_{i=1}^n \partial_i I = I(x, y)$$

## F. Color Space Choice & Optimization

The question arises of how a colored image may be pre-distorted. The outline presented in I.A.1 assumes that  $I(x, y)$  is a monocolor image, with  $(x, y)$  coordinates corresponding to a single intensity. Color images are typically split into color channels. Often, R (red), G (green), and B (blue) color channels are used. The original image may be represented in the form:

$$I(x, y) = (r(x, y), g(x, y), b(x, y)),$$

And we may thus apply convolution or deconvolution operations to the individual color channel functions  $r, g, b : \mathbb{Z}^3 \rightarrow \text{clr}$ , where  $\text{clr} := \{\mu \mid 0 \leq \mu \leq 255; \mu \in \mathbb{Z}\}$

Deconvolution is a space-intensive process. The fast fourier transform alone is of space complexity  $\mathcal{O}(n \log n)$ , and the deconvolution operation's complexity is dominated by the fourier transform. It is thus infeasible to conduct deconvolution operations on each of the  $r, g, b$  channels, which triples the processing time and takes a high computing toll. Additionally, we observe severe color bleeding and color perversion in using  $r, g, b$  separate channel filtering. In the case of  $r, g, b$  channel deconvolution, the deconvolution is represented:

$$(r(x, y), g(x, y), b(x, y)) \rightarrow \quad (5)$$

$$(r(x, y) * h(x, y), g(x, y) * h(x, y), b(x, y) * h(x, y)). \quad (6)$$

That is, three deconvolutions must take place; one for each of the channels. One may, instead, attempt to decompose the full-color image into a different set of channels,

representative of different aspects of the color. For instance, YUV (in modern displays, this is YCbCr). Y is the luma channel—solely the luma channel will give a grayscale version of the image. U and V represent deficiency in red and blue respectively. Our decomposition becomes:

$$I(x, y) = (y(x, y), u(x, y), v(x, y)),$$

and we see that only the luma channel must be pre-distorted. Non-analytically, the U and V channels will retain their original colors, and the eye will perceive the colors to fit into the shapes defined in the luma channel. We note that  $y, u, v : \mathbb{Z}^3 \rightarrow \text{clr}$ ; here,  $\text{clr} := \{\mu \mid 0 \leq \mu \leq 1; \mu \in \mathbb{R}\}$  as a matter of convention. Now, the deconvolution is given:

$$(y(x, y), u(x, y), v(x, y)) \rightarrow \quad (7)$$

$$(y(x, y) * h(x, y), u(x, y), v(x, y)), \quad (8)$$

reducing the operation to a single deconvolution. We may alternatively use the LMS color space, which decomposes colors in a similar manner (one luma and two chroma channels), but while taking into account the human eye's perception to different color intensities. We notice a very subtle difference in the simulation, between the two color spaces.

In integrating this into the live vision correction, we use graphics pipelining to process multiple images simultaneously to avoid lag. Additionally, individual images were segmented into smaller tiles, that were then processed. As the complexity of deconvolution is of a greater order than linear, processing multiple smaller images takes less time than processing a single large image.

## G. Live User Location Tracking

To ensure real-time efficacy for users of the vision-correcting display, it is imperative that the device functions optimally across diverse viewing angles. Achieving the pre-distortion of the PSF can be effectively realized through a perspective transformation, by the following steps:

1. Normalize the input image to a grayscale format and scale pixel values to the range  $[0, 255]$ .

2. Represent 2D image points as homogeneous 3D coordinates.
3. Define the matrix for projection. The third column of the projection matrix translates the coordinates, shifting the origin to the center of the image

$$P = \begin{bmatrix} 1 & 0 & -\frac{c}{2} \\ 0 & 1 & -\frac{r}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4. Define the matrix for rotation around the  $x$ - and  $y$ -axes ( $R_x$  and  $R_y$ ). The former simulates the effect of tilting or rotating the camera or viewing perspective along the horizontal axis. The rotation is defined by angle ( $\theta_x$ ). The off-diagonal elements  $-\sin(\theta_x)$  and  $\sin(\theta_x)$  introduce the rotation effect. The cosine terms  $\cos(\theta_x)$  preserve spatial relationships during the rotation. As ( $\theta_x$ ) increases, the cosine term determines how much of the original y-coordinate is preserved during the rotation. This is similar in the  $R_y$  vertical tilt matrix as well.

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Define a matrix  $T$  for the translation along the  $z$ -axis—this corresponds to moving the camera or the viewpoint forward or backward. The translation is applied to the homogeneous coordinate, introducing a shift along the depth ( $z$ ) axis. The third column ensures no change in the  $z$ -coordinate, except for the translation along the  $z$ -axis represented by  $\xi$ . The matrix is as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \xi \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6. Define the camera intrinsics matrix  $A_2$ . Here,  $f$  refers to focal length, determined based on the desired field of view. The principal point on the  $x$ - and  $y$ -axes is defined by the midpoint. The third row is fixed for homogeneous coordinates.  $A_2$  converts 3D coordinates to 2D image coordinates, incorporating camera intrinsics. The FOV used in the prototype is  $80^\circ$ —the larger the FOV, the wider the perspective captured.  $c, r$  are the columns and rows.

$$A_2 = \begin{bmatrix} f & 0 & \frac{c}{2} & 0 \\ 0 & f & \frac{r}{2} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

7. The matrices are multiplied in this order to achieve the desired perspective transformation. Transformation of the image by the perspective transformation matrix results in the pre-distorted output PSF, adapting the image to different viewing angles. Finally, the warped image is normalized, ensuring pixel values are within a valid intensity range.

$$(A_2 T R_y R_x A_1) K = K_t,$$

where  $K$  is the discretized PSF  $k(x, y)$  in matrix form.

The algorithm for live calculation of the user's distance from the screen involves an infinite loop, wherein each frame is captured, converted to RGB, and face locations are identified using dlib's face recognition model. The model used has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

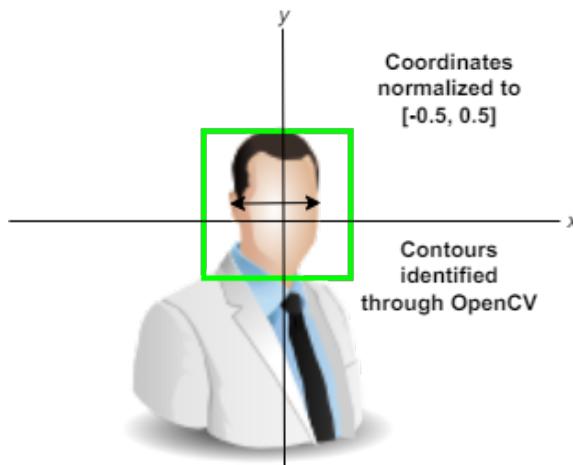


FIG. 2. Representation of face recognition implementation

The width of the face in pixels is set as a constant value. Then, using the horizontal FOV of the camera, the distance is calculated based on the principle that the user's face and its image on the live feed form similar triangles.

$$r = \frac{u_w f_w}{2f_p \tan\left(\frac{\text{FOV}_h}{2}\right)},$$

where  $r$  represents the absolute distance,  $u_w$  represents the user's face width in cm,  $f_w$  represents the width of the frame,  $f_p$  represents the face width in pixels as it appears on the live feed, and  $\text{FOV}_h$  represents the horizontal FOV of the camera.

The calculated distance  $r$  represents the estimated distance of the user from the screen in the given frame. It is to be acknowledged that the distance calculation may deviate from the actual distance. If advanced depth sensing technologies such as Time-of-Flight (ToF) cameras are used, where the time taken for emitted light to travel to the user and back is measured, precise information can be gathered. This would surpass traditional methods by mitigating the impact of ambient lighting conditions and ensuring robust performance across diverse environments. An alternative approach involves utilizing multiple sensors in a stereo vision setup enhances accuracy further by triangulating the user's position. It is worth noting that such advanced setups require sophisticated hardware and computational resources, and does not meet the criteria for a live display that the layman should be able to use.

The algorithm for live calculation of the user's horizontal and vertical angles occurs simultaneously. Within the algorithm, the user's face is localized in each frame, and the horizontal and vertical angles are computed based on the face's midpoint position within the frame. The horizontal angle is calculated by normalizing the face's midpoint deviation from the center of the frame, while the vertical angle is similarly determined using the midpoint's deviation from the frame's vertical center. These angles are then converted from degrees to radians, and the resulting data is stored.

## H. Tiling and Optimization

The Wiener deconvolution algorithm runs with  $\mathcal{O}(n \log n)$  complexity, where  $n$  are the data size. The deconvolution algorithm's time complexity is dominated by the Fast

Fourier Transform (FFT) and Inverse FFT (IFFT) operations used in its computation.

By slicing the original image into tiles, the 2D data size  $n$  is quadratically reduced, resulting in a much lower time complexity per individual tile (e.g., for division into tiles as powers of 2,  $\mathcal{O}\left(\frac{n}{2^a} \log \frac{n}{2^a}\right)$ ). For the total number of tiles, the order of complexity becomes  $\mathcal{O}\left(n \log \frac{n}{2^a}\right)$ , which is much lower than the original time complexity. The tiles are then easily stitched back together—for instance, using basic array manipulation.

### I. Higher Order Aberrations

Hartmann-Shack Wavefront Aberrometers give an empirical reading of the magnitude of different refractive aberrations—this can be used to reconstruct a PSF. As seen in the subsequent figures, the deconvolution algorithm also works for such non-circular PSFs. However, the tiling method described above does not work for these, as they are not continuously rotationally symmetrical.

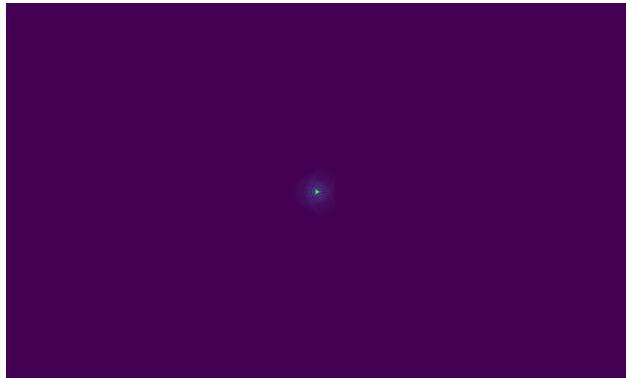


FIG. 3. Oblique trefoil at -1.5D, 500mm from screen.

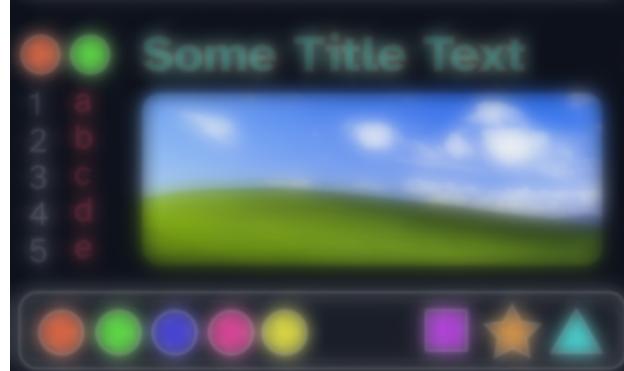


FIG. 4. Non-trefoil affected user User's perception of image without correction.



FIG. 5. Perceived image for trefoil-afflicted user.

#### IV. RESULTS

To assess the performance of our display, we compare the original image with the reconvolved image, which simulates how the corrected image appears to the viewer. We evaluate the quality of the reconvolved image using multiple metrics: Peak Signal-to-Noise Ratio (PSNR), Root Mean Square Error (RMSE), Absolute Error (AE), Normalized Cross-Correlation (NCC), and Structural Similarity Index (SSIM). These metrics provide insights into the effectiveness of our approach in mitigating visual aberrations. The following images will be utilized for comparison.



FIG. 6. Original (reference) image

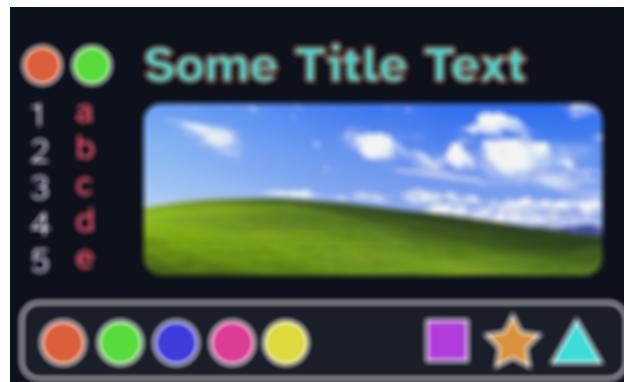


FIG. 7. Blurred (perceived) image without correction

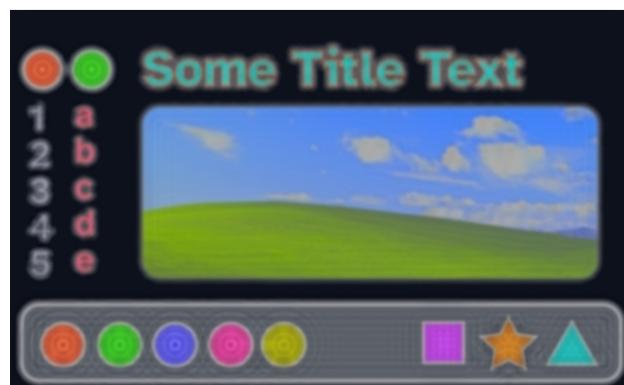


FIG. 8. Reconvolved (perceived) image with correction

PSNR is a widely used metric for evaluating image quality by comparing the original and distorted images. It measures the ratio between the maximum possible signal

value and the magnitude of noise introduced during image reconstruction. Higher PSNR values indicate better fidelity, implying that the reconolved image closely resembles the original [9]. It is given:

$$\text{PSNR} = 10 \log \left( \frac{\max(I(x, y))^2}{\overline{E_{\text{sq}}}} \right) \quad (9)$$

where  $I$  is the image, and  $\overline{E_{\text{sq}}}$  represents the mean square error between the original and reconolved images.

The PSNR was computed across different colour channels and collectively for the entire image. Specifically, the overall PSNR value stands at 14.3268 dB, with minimal variation across the blue, green, and red channels - 14.4072 dB, 14.3667 dB, and 14.2091 dB respectively. These values signify a moderate level of noise in the reconolved image, suggesting areas for optimization in the deconvolution parameters to achieve a closer approximation to the original image quality. The consistency across channels also confirms that our approach maintains colour integrity, which is crucial for practical applications.

The Absolute Error (AE) measures the discrepancy between the corresponding pixel values of the original and reconolved images. It is calculated as the sum of the absolute differences between the pixel intensities of the two images [10]. AE for each pixel is defined as:

$$E_{\text{abs}} = \mathbb{m}(I(x, y)) \ominus \mathbb{m}([I(x, y) * h(x, y)] * k(x, y)), \quad (10)$$

where  $I$  is the original image,  $\mathbb{m} : \mathbb{Z}^3 \rightarrow \mathbb{Z}$  is the monochromatic filter function. As usual  $k, h$  respectively denote the PSF and IPSF. Here,  $\cdot \ominus \cdot$  is naturally a component-wise subtraction operation.

AE was computed separately for each color channel as well as collectively for the entire image. This showed a level of variance in pixel values across all channels, with a percentage difference of 10.67% from the original image. However, this is not a demonstration of perceived clarity, and is simply a pixel-wise difference.



FIG. 9. Highlighting pixel differences. It can be seen that differences primarily exist at the edges of the image.

The Structural Similarity Index (SSIM) is a metric used to assess the perceived quality of digital images and videos. SSIM is considered superior to simpler metrics such as Mean Squared Error (MSE) and PSNR because it incorporates perceptual phenomena, including texture, luminance, and contrast, which are critical in human visual perception [11]. SSIM considers the closeness of the image luminance, which reflects the average brightness, it evaluates the similarity in contrast, which involves measuring the standard deviation of the pixel intensities, and it assesses the correlation coefficient between the two images, which quantifies how structural elements in the images correlate [12]. It is given:

$$\text{ssim}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (11)$$

where  $\mu_x$  and  $\mu_y$  represent the mean intensity values of images  $x$  and  $y$ ,  $\sigma_x$  and  $\sigma_y$  are the standard deviations, and  $\sigma_{xy}$  is the covariance of the two images. The constants  $c_1$  and  $c_2$  are included to stabilize the division with a weak denominator.

In fig. 6 and fig. 8, the SSIM value achieved was 83.04%, indicating a high degree of similarity between the original and the corrected images viewed through the display.

	Our Work	Huang et. al, 2014 [4]	Huang et. al, 2012 [13]	Pamplona et. al, 2012 [14]
SSIM	83.04%	50.59%	27.89%	47.28%
PSNR (RGB)	14.3268	15.1538	8.4279	15.1366

PSNR (% diff.)	0.00%	0.03%	0.02%	0.03%
NCC (Coefficient)	0.742	0.423	0.206	0.380
Contrast	100%	45%	15%	100%

TABLE I: Comparison of our work with past research using the aforementioned metrics.

## V. CONCLUSION

### A. Limitations

The deconvolution approach does not work well for small images due to their rasterized nature—thus, the deconvolution should ideally be performed on a continuous vectorized image, as at small scales, ringing artefacts tend to dominate the image, obscuring relevant details.

We also note the presence of seams where the image has been cut during the tiling process—their presence is apparent only because of the ringing artefacts. To avoid this, the image may be broken up into tiles with some padding, so that the ringing artefacts around the edges do not reduce clarity.

Similarly, the tiling approach described does not work for higher-order aberrations, because the initial aberration is applied to the entire image, and may involve shifting elements from one tile to another, resulting in a misaligned reconolved image. This may be avoided, and the processing time may be reduced significantly, if tiling is not used—rather, the image may be broken up based on the presence of contours around letters and images. These can later be individually overlayed onto an undistorted background instead of separately tiling and masking.

## B. Summary and Improvements

Thus, we see that our work has enabled approximate vision correction for refractive aberrations (SSIM = 83.04%). This can be made considerably more accurate for the end user by allowing for empirical PSF input from a wavefront aberrometer. We have successfully optimized the traditional deconvolution approach heavily for the real-time vision correction aim, making use of image tiling and parallel processing, whilst contributing to signal processing through YUV space deconvolution—this can be used in traditional deconvolution applications as well. These optimizations and novel contributions are a step forward in making vision correcting displays feasible solutions in the real world.

Improvements to this implementation could include the creation of a parallax barrier to separate left and right eye perception to accommodate different left and right power.

## ACKNOWLEDGMENTS

We thank Prof. Partha Pratim Das for his invaluable counsel and guidance in this project. His expertise in image processing helped us refine our implementation and increase its applicability. We are immensely grateful to Ashoka University and Mphasis Limited for giving us the opportunity to carry forward our work and for offering us the mentorship without which this project would not have been possible.

- 
- [1] J. D. Steinmetz, R. R. Bourne, P. S. Briant, S. R. Flaxman, H. R. Taylor, J. B. Jonas, A. A. Abdoli, W. A. Abrha, A. Abualhasan, E. G. Abu-Gharbieh, *et al.*, “Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study,” *The Lancet Global Health*, vol. 9, no. 2, pp. e144–e160, 2021.
  - [2] B. A. Holden, T. R. Fricke, D. A. Wilson, M. Jong, K. S. Naidoo, P. Sankaridurg, T. Y. Wong, T. J. Naduvilath, and S. Resnikoff, “Global prevalence of myopia and high myopia and temporal trends from 2000 through 2050,” *Ophthalmology*, vol. 123, no. 5, pp. 1036–1042, 2016.
  - [3] M. Markoulli and S. Kolanu, “Contact lens wear and dry eyes: challenges and solutions,” *Clinical optometry*, vol. 9, p. 41, 2017.
  - [4] F.-C. Huang, G. Wetzstein, B. A. Barsky, and R. Raskar, “Eyeglasses-free display: Towards correcting visual aberrations with computational light field displays,” *ACM Trans. Graph.*, vol. 33, 7 2014.
  - [5] D. Kim, S.-W. Nam, K. Bang, B. Lee, S. Lee, Y. Jeong, J.-M. Seo, and B. Lee, “Vision-correcting holographic display: evaluation of aberration correcting hologram,” *Biomedical Optics Express*, vol. 12, no. 8, pp. 5179–5195, 2021.
  - [6] G. Xue, J. Liu, X. Li, J. Jia, Z. Zhang, B. Hu, and Y. Wang, “Multiplexing encoding method for full-color dynamic 3d holographic display,” *Optics express*, vol. 22, no. 15, pp. 18473–18482, 2014.
  - [7] P.-A. Blanche, A. Bablumian, R. Voorakaranam, C. Christenson, W. Lin, T. Gu, D. Flores, P. Wang, W.-Y. Hsieh, M. Kathaperumal, *et al.*, “Holographic three-dimensional telepresence using large-area photorefractive polymer,” *Nature*, vol. 468, no. 7320, pp. 80–83, 2010.
  - [8] M. Alonso Jr, A. Barreto, and J. G. Cremades, “Image pre-compensation to facilitate computer access for users with refractive errors,” *ACM SIGACCESS Accessibility and Computing*, no. 77-78, pp. 126–132, 2003.
  - [9] A. Hore and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *2010 20th international*

- conference on pattern recognition*, pp. 2366–2369, IEEE, 2010.
- [10] T. O. Hodson, “Root mean square error (rmse) or mean absolute error (mae): When to use them or not,” *Geoscientific Model Development Discussions*, vol. 2022, pp. 1–10, 2022.
- [11] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through fsim, ssim, mse and psnr—a comparative study,” *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.
- [12] P. Ndajah, H. Kikuchi, M. Yukawa, H. Watanabe, and S. Muramatsu, “Ssim image quality metric for denoised images,” in *Proc. 3rd WSEAS Int. Conf. on Visualization, Imaging and Simulation*, pp. 53–58, 2010.
- [13] F.-C. Huang, D. Lanman, B. A. Barsky, and R. Raskar, “Correcting for optical aberrations using multilayer displays,” *ACM transactions on graphics (TOG)*, vol. 31, no. 6, pp. 1–12, 2012.
- [14] V. F. Pamplona, M. M. Oliveira, D. G. Aliaga, and R. Raskar, “Tailored displays to compensate for visual aberrations,” *ACM transactions on graphics (TOG)*, vol. 31, no. 4, pp. 1–12, 2012.
- [15] C. Reuben and N. Elgaddal, “Quickstats: Percentage of children aged 2–17 years who wear glasses or contact lenses, by sex and age group—national health interview survey, united states, 2019,” 2021.
- [16] B. A. Barsky, F.-C. Huang, D. Lanman, G. Wetzstein, and R. Raskar, “Vision correcting displays based on inverse blurring and aberration compensation,” in *Computer Vision - ECCV 2014 Workshops* (L. Agapito, M. M. Bronstein, and C. Rother, eds.), (Cham), pp. 524–538, Springer International Publishing, 2015.
- [17] M. L. Rubin, “Spectacles: past, present, and future,” *Survey of Ophthalmology*, vol. 30, no. 5, pp. 321–327, 1986.
- [18] R. Rashed, “Ibn al-haytham (alhazen),” *Encyclopaedia of the history of science, technology, and medicine in non-western cultures. Berlin: Springer-Verlag*, pp. 1090–1093, 2008.
- [19] L. Szczotka-Flynn, D. G. Ahearn, J. Barr, W. J. Benjamin, T. Kiang, J. J. Nichols, O. D. Schein, R. P. Stone, and L. Winterton, “1. history, evolution, and evolving standards of contact lens care,” *Contact Lens and Anterior Eye*, vol. 36, pp. S4–S8, 2013.
- [20] L. J. Kugler and M. X. Wang, “Lasers in refractive surgery: history, present, and future,” *Applied optics*, vol. 49, no. 25, pp. F1–F9, 2010.

- [21] F.-C. Huang, *A Computational Light Field Display for Correcting Visual Aberrations*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2013.
- [22] I. Patel and S. West, “Presbyopia: prevalence, impact, and interventions.,” *Community eye health*, vol. 20, no. 63, pp. 40–41, 2007.
- [23] R. C. Gonzalez and R. E. Woods, “Digital image processing,” *Reading, Ma*, 1992.
- [24] D. Fish, A. Brinicombe, E. Pike, and J. Walker, “Blind deconvolution by means of the richardson–lucy algorithm,” *JOSA A*, vol. 12, no. 1, pp. 58–65, 1995.
- [25] W. H. Richardson, “Bayesian-based iterative method of image restoration,” *JoSA*, vol. 62, no. 1, pp. 55–59, 1972.
- [26] A. Agrawal, Y. Xu, and R. Raskar, “Invertible motion blur in video,” in *ACM SIGGRAPH 2009 papers*, pp. 1–8, 2009.
- [27] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, “Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing,” *ACM Trans. Graph.*, vol. 26, no. 3, p. 69, 2007.
- [28] J. R. Bergen and E. H. Adelson, “The plenoptic function and the elements of early vision,” *Computational models of visual processing*, vol. 1, p. 8, 1991.
- [29] H. Strasburger, M. Bach, and S. Heinrich, “Blur unblurred—a mini tutorial,” *i-Perception*, vol. 9, p. 204166951876585, 03 2018.
- [30] C. Ding, *Algorithms and applications of the vision correcting display*. PhD thesis, MA thesis. EECS Department, University of California, Berkeley, 2017.
- [31] K. Akeley, S. J. Watt, A. R. Girshick, and M. S. Banks, “A stereo display prototype with multiple focal distances,” *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 804–813, 2004.
- [32] R. Lane and M. Tallon, “Wave-front reconstruction using a shack–hartmann sensor,” *Applied optics*, vol. 31, no. 32, pp. 6902–6908, 1992.
- [33] V. F. Pamplona, A. Mohan, M. M. Oliveira, and R. Raskar, “Netra: interactive display for estimating refractive errors and focal range,” in *ACM SIGGRAPH 2010 papers*, pp. 1–8, 2010.
- [34] F. Laricchia, “Display technology | statista.” <https://www.statista.com/topics/3288/display-technology/>, 2 2022. (Accessed on 12/02/2022).
- [35] M. Alonso and A. Barreto, “Pre-compensation for high-order aberrations of the hu-

man eye using on-screen image deconvolution," *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 1, pp. 556–559 Vol.1, 2003.