Ghostbuster: A Software Approach for Reducing Ghosting Effect on Electrophoretic Displays

Tao Hu, *Member, IEEE*, Menglong Cui, Mingsong Lv, *Member, IEEE*, Tao Yang, Yiyang Zhou, Qingxu Deng, *Senior Member, IEEE*, Chun Jason Xue, *Member, IEEE*, and Nan Guan, *Member, IEEE*

***Abstract*—Electrophoretic displays (EPDs), also known as e-paper, offer a paper-like visual experience by reflecting ambient light, making them distinct from traditional LCD or LED displays. They are favored for their eye comfort, energy efficiency, and material flexibility, which make them appealing for a wide range of embedded devices, including eReaders, smartphones, tablets, and wearables. However, EPDs face a significant challenge: the necessity for a fast refresh rate (to maintain an acceptable display performance) introduces a pronounced ghosting effect. This effect results in noticeable color discrepancies between the displayed and source images, harming the user experience and hindering EPDs' broader application in devices requiring dynamic content display. This article proposes a software-based solution to address the ghosting issue in EPDs. Our approach involves developing analytical models to predict the occurrence of ghosting effects and adjusting the source images to counteract the anticipated color deviations, which can reduce the perceivable ghosts on the display. Experimental evaluation conducted on real-world EPDs validates the effectiveness of our proposed approach in reducing the ghosting effect.**

***Index Terms*—E ink, electrophoretic display, embedded software, ghosting effect reduction.**

I. Introduction

ELECTROPHORETIC displays (EPDs), also known as electronic paper or e-paper, emerge as a distinctive class of display technology that mimics the appearance of ink on paper by reflecting ambient light, providing a comfortable reading experience akin to traditional paper [1]. EPDs offer distinct advantages over alternative display technologies, making them particularly appealing for embedded devices.

One standout benefit of EPDs is their commitment to ocular health. In contrast to emissive displays, such as LCDs, LEDs, and OLEDs, which generate blue light contributing to eye strain and potential long-term retinal damage [2], [3], [4], EPDs reflect natural light, thereby reducing blue light exposure and its associated risks. This characteristic positions EPDs as a safer alternative for prolonged screen usage. Moreover, EPDs are recognized for their energy conservation, drawing power only during the refresh phase and capable of maintaining an image without a continuous power supply. The construction of EPDs, involving an ink layer laminated to a plastic film substrate, also lends to their flexibility, durability, and lightweight design. Over the past decade, the inherent paper-like quality of EPDs has led to their widespread adoption in eReaders [5]. More recently, their application has extended to a variety of embedded devices, including wearables [6], [7], [8], [9], smartphones [10], [11], [12], tablets [13], [14], [15], [16], [17], laptops [18], [19], [20], and even desktop monitors [21], [22], [23], reflecting their growing versatility and appeal in the tech market.

Despite these advantages, the adoption of EPDs beyond eReaders has been hindered by challenges associated with dynamic content display, primarily due to prolonged refresh times and noticeable screen flickering. These challenges root in the composition of EPDs, where reflective particles manipulated by time-varying electric fields (known as waveforms) exhibit specific grayscale colors [24], [25]. To ensure color precision, the waveforms contain an activation phase for precise particle control, flipping the whole screen between black and white, which incurs a long time delay (typically up to 1 s) and screen flickering. Although these issues are tolerable on eReaders where screen updates are infrequent, they become unacceptable for applications requiring higher refresh rates, such as Web browsing and video playback.

To address these limitations, EPD manufacturers have introduced a *fast refresh* mode that employs short waveforms [1] by skipping the activation phase to shorten the refresh process. While this technique reduces refresh delays (around 1/5 the time required by complete waveforms) and removes flickering, it introduces a significant drawback: the *ghosting effect*. This phenomenon results from insufficient control over particle movement, leading to residual images or "ghosts"

Manuscript received 12 August 2024; accepted 13 August 2024. This work was supported in part by the Huawei Innovation Research Program "Energy-Efficiency Aware Collaborative Computing for Heterogeneous OS Platforms"; in part by the Hong Kong GRF under Grant 15206221, Grant 11209122, and Grant 11208522; and in part by the National Natural Science Foundation of China under Grant 61772123 and Grant 62072085. This article was presented at the International Conference on Embedded Software (EMSOFT) 2024 and appeared as part of the ESWEEK-TCAD special issue. This article was recommended by Associate Editor S. Dailey. *(Corresponding author: Mingsong Lv.)*

Tao Hu is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China.

Menglong Cui is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

Mingsong Lv is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and also with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: mingsong.lyu@polyu.edu.hk).

Tao Yang and Yiyang Zhou are with the 2012 Labs, Central Software Institute, Huawei Technologies Company Ltd., Beijing 100015, China.

Qingxu Deng is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China.

Chun Jason Xue is with the Department of Computer Science, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.

Nan Guan is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Digital Object Identifier 10.1109/TCAD.2024.3446711

1937-4151 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

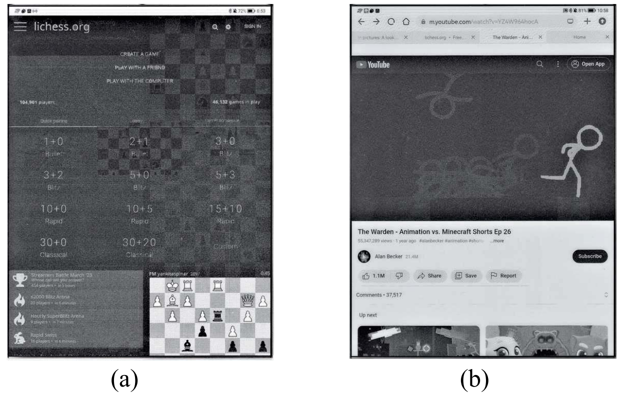


Fig. 1. Examples of the ghosting effect on EPDs. (a) As the screen content moves downward, the chess board leaves ghost images in the areas it passed (where a single-color dark gray is expected). (b) In the video, a stick figure moves in consecutive frames. The figure's moving history are left in the dark background as a result of ghosting.

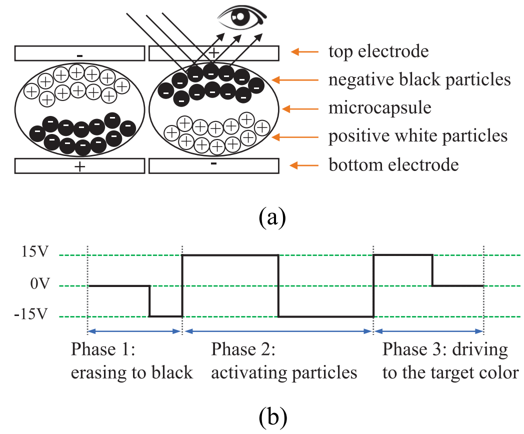


Fig. 2. Structure and driving waveform of microcapsuled EPDs [25]. (a) Physical structure of EPDs. (b) Waveform with an activation phase.

from previous frames (as shown in Fig. 1), compromising screen readability and user experience.

Hardware-based techniques have been developed to address this issue by carefully tuning the short waveforms. However, it can only alleviate the ghosting effect to a limited extent due to the inherent limitation of the short waveforms. Fig. 1 shows the screenshots on a state-of-the-art EPD embedded device with carefully tuned short waveforms, where ghosts are still noticeable on the screen.

In light of the limitations of hardware-based solutions, this article proposes a *software-based* approach to mitigate the ghosting effect on EPDs. The main idea of our approach is to modify the source image to be displayed to counteract the color deviations induced by ghosting when the source image is updated over previous images on the EPD device. A key challenge is the accurate prediction of ghosting occurrence. To address this, we built analytical models to characterize the ghosting effects and the resultant ghost images from a thorough exploration of the screen update history. Built upon these models, we adjust the colors in the original image to mitigate the ghosting effects. To the best of our knowledge, this represents the first attempt at employing a software-based solution to tackle the ghosting effect in EPDs. Experiments on real-world EPD devices have validated the effectiveness of our approach, suggesting it as a viable path toward increasing the utility and adoption of EPDs in a variety of embedded devices.

II. Preliminary

A. EPD Basics

Basic Structure: EPDs are reflective display devices [25]. The fundamental structure of an EPD consists of microcapsules containing positively charged white particles and negatively charged black particles suspended in a clear fluid [Fig. 2(a)]. Positioned between two transparent electrodes, the application of an electrical voltage causes the movement of these particles, thereby altering the displayed grayscale colors visible to the viewer.[1] For instance, applying a positive voltage to the bottom electrode drives white particles to move upward, displaying the white color. Inverting the voltage yields the black color. EPDs employ an active matrix for pixel-by-pixel display control, similar to the technology used in LCDs. However, the electric fields from adjacent pixels may influence the particle movement at their boundaries.

Waveform: The multitude of black and white particles in microcapsules exhibit highly nonlinear responses to electric fields. To precisely position these particles for accurate grayscale display, EPDs employ a *waveform*, a time-variant electric field applied to each pixel. This waveform includes an *activation* phase, or *particle shaking* phase, characterized by a series of voltage alternations designed to "loosen" the particles for subsequent accurate placement [Fig. 2(b)]. Despite its effectiveness in particle control, the full duration of the waveform, often extending up to around 1500 ms, introduces a flickering effect between black and white states, which can be particularly disruptive during fast content updates.

Fast Refresh: EPDs address the flickering problems by *fast refresh*, using *short waveforms* that bypass the activation phase to drive the display. Although fast refresh effectively removes flickering and accelerates the refresh process (often to just 1/5 of that required by a full waveform), it weakens particle control, leading to notable deviations in grayscale accuracy and, consequently, a degraded display quality.

Dithering: The weak particle control under fast refresh mode necessitates a binary approach to pixel representation, confining the display capabilities to black and white. To simulate intermediate grayscale colors under this constraint, a *dithering* process is integrated into the display update pipeline, transforming the original image into a new image with exclusively black and white pixels. Using algorithms like Floyd–Steinberg [27] to produce different distributions of black and white pixels allows the intended grayscale colors to be visually represented (Fig. 3). "Fast refresh + dithering" is currently a standard option on EPDs—particularly suitable for applications demanding frequent screen updates.

[1] Although colorful EPDs also exist [26], they are rarely used in mobile devices requiring fast refresh. In this work, we focus on grayscale EPDs.

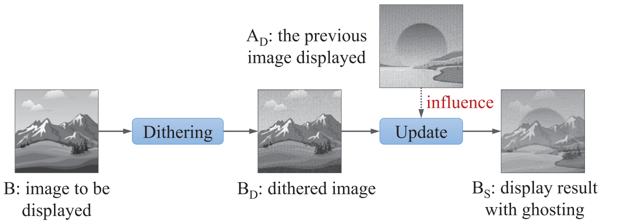


Fig. 3. Standard image update pipeline in EPD fast refresh mode and how ghosting effect has resulted.

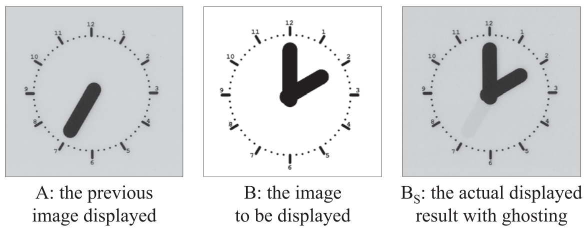


Fig. 4. Example of ghosting due to misdriving.

B. Ghosting Effects

The fast refresh mode's reliance on short waveforms compromises the accurate display of intermediate grayscales, leading to pixel color deviations. These deviations, influenced by previously displayed images, manifest as the *ghosting effect*, where the discernible residual image of the prior frame is visible in the new frame. Fig. 3 exemplifies this with a ghost image of a sun from a preceding frame, affecting the faithfulness in the display of the subsequent frame. Despite extensive efforts to refine short waveforms through hardware adjustments, the inherent constraints of these waveforms mean that completely eliminating ghosting remains impossible. So far, the ghosting effect persists as a significant issue, adversely impacting display quality. Two ghosting effects, namely, *misdriving* and *fringing*, are most commonly observed in microcapsuled EPD devices and are frequently reported in the related literature.

Misdriving: This issue arises from inadequate control over particle positioning, resulting in most particles within a pixel failing to reach their intended locations and causing color inaccuracies [1]. This phenomenon is particularly pronounced when a pixel displays opposite colors in consecutive images, as illustrated in Fig. 4. The display begins with the EPD presenting image A and, subsequently, image B. The pixels in the clock hand, pointing to 7 in image A, undergo a transition from black to white. Due to misdriving, a distinctive gray shade is left in the original position of the clock hand.

Fringing: The compact size of microcapsules relative to pixels means that some capsules astride the boundaries between adjacent pixels. The application of waveforms can inadvertently extend the electric field from one pixel to its neighbor, influencing microcapsules at the border, known as *fringing* or *crosstalk* [28], [29]. The intensity of fringing correlates with the electric field's strength and is more pronounced with the use of short waveforms, leading to visible color deviations at pixel edges (depicted in Fig. 5).

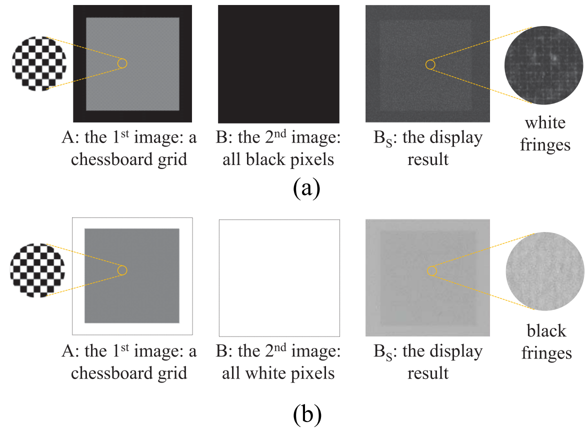


Fig. 5. Example of ghosting due to fringing. (a) Ghosting due to crosstalk—white fringes. (b) Ghosting due to crosstalk—black fringes.

EPD devices working with slow refresh modes (driven by long waveforms with activation phases) generally do not exhibit the ghosting effect problem. Many EPD-based mobile devices also feature a "reset" operation to restore pixels to their original colors. This is achieved by initiating a slow refresh cycle that includes an activation phase. The "reset" can be activated manually by the user or automatically by the device at intervals. However, this slow refresh "reset" introduces screen flickering, which can be problematic when displaying dynamic content.

III. Overview of Our Approach

Given the limitations of hardware solutions in addressing the ghosting effect in EPDs, this article introduces a software-based approach to mitigate this issue. The fundamental concept involves modifying the source image to be displayed, aiming to compensate for the color deviations caused by ghost images. An example of this is depicted in Fig. 6, where two consecutive image frames are sent to an EPD device. The first frame includes an image of the sun, which, when updated with the second frame, leaves a residual ghost of the sun on the display. To address this, our method modifies the second frame by adjusting the colors in the region where the sun's ghost would appear. This adjustment is intended to counterbalance the ghosting effect from the first frame, thereby reducing the ghosting effect perceived by users. To realize this, we need to perform two tasks.

1) *Ghosting Effect Modeling:*The initial step requires an accurate prediction of the ghosting effects that might occur when the new image is overlaid on the previous one on the EPD device. We have developed a method to model the ghosting effect by analyzing the dithered images of the 1st and the 2nd frames. This model predicts the ghosting effect for each pixel (the analysis result).
2) *Ghosting Effect Reduction:* With the predicted ghosting information from the modeling step, we then apply image processing techniques to the source image (the original image of the 2nd frame). The goal is to alter the colors of pixels prone to ghosting, aiming for these

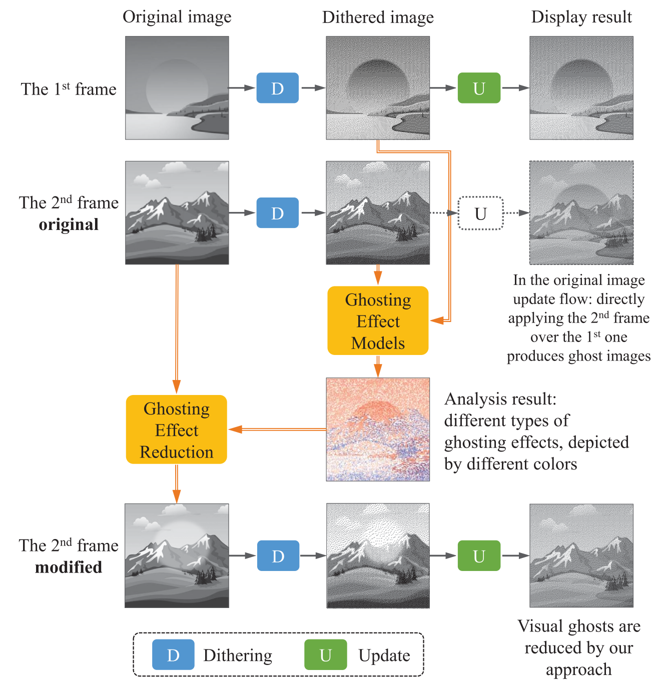


Fig. 6. Overview of our approach.

changes to counteract the predicted ghosting effects (in the modified image of the 2nd frame).

Fig. 6 shows the new image update workflow. Note that the dotted "U" step refers to the original image update workflow without our approach, by which ghosts can be produced on the EPD screen. The details of the two tasks will be further discussed in Sections IV and V, respectively.

IV. Ghosting Effect Modeling

The process of ghosting effect modeling begins by analyzing the update histories of individual pixels. This analysis leads to the development of an analytical model capable of predicting the resultant color on the EPD screen for a pixel, given its specific update history.

Our approach distinguishes between two primary types of ghosting effects, misdriving and fringing, due to their distinct manifestations within the operational characteristics of EPD devices. Misdriving influences the central color of a pixel, where the displayed color might be influenced by one or several of the most recent updates, an effect occurring in the time domain. On the other hand, fringing primarily occurs in the spatial domain, affecting the edges of a pixel. This effect arises when adjacent pixels undergo specific color update patterns, leading to unintended colors appearing at the pixel boundaries. By treating misdriving and fringing as separate issues, our model effectively explores the distinct dimensions, time and space, in which the ghosting effects operate. This separation allows for a more faithful understanding and prediction of the ghosting phenomena.

In the following, we develop analytical models describing the two ghosting effects. Recognizing that these ghosting phenomena affect separate regions of a pixel, we proceed with a color calibration procedure designed to empirically ascertain the display outcomes under the ghosting scenarios. The color deviation of the pixel that experiences ghosting effects can then be computed and used in ghosting effect reduction.

A. Definitions of Colors

Before presenting our modeling approach, it is essential to first clarify the terminology related to "colors" as used within the context of this article, particularly since our examination is centered around grayscale EPDs. In these displays, the color of each pixel in the source image is determined by a grayscale value, which spans from 0 (representing black) to 255 (representing white).

It is important to recognize that there often exists a discrepancy between the intended color and the color that is actually rendered on the display, a phenomenon that is not unique to EPDs but common across various display technologies. In our discussions, when we mention driving or updating a pixel to black or white, we refer to the original color specified in the source image. Conversely, "displayed colors" refer to the colors that are visually perceived on the EPD screen. For the sake of clarity, we adopt the notations B and W to denote the grayscale extremes of 0 and 255, respectively, and utilize $\widetilde{B}$ and $\widetilde{W}$ to indicate the actual rendered grayscale levels when a pixel is driven to black or white (without being affected by ghosting effects). To obtain the grayscale value of the displayed colors, we scan the EPD screen and use the grayscale values read from the scanned images.

Moreover, considering the potential nonuniformity in a pixel's color due to the fringing effects, we define a pixel's color based on its average color. This average considers both the pixel's central area and its fringes. Similarly, when discussing the color of a larger region encompassing multiple pixels, we refer to the mean grayscale value derived from all the pixels within that specific area.

B. Modeling Misdriving

The task of accurately modeling misdriving is challenging, primarily due to the nonlinear response of the particles in EPD devices to the applied waveforms and the influence of a pixel's update history on its displayed color. To address these complexities, our approach involves an in-depth exploration and analysis of the display outcomes following various update history sequences, aiming to identify patterns that can be expressed by analytical models. Our discussion here focuses on individual pixels, with the understanding that the findings are applicable across all the pixels on the EPD screen.

Our methodology entails an exhaustive enumeration of possible update histories for a pixel, considering a defined number of consecutive updates (the methodology for efficiently conducting this exploration in parallel is detailed in Section VI). We limit our exploration to update sequences up to seven steps long, encompassing up to 128 distinct histories. The experimentation was conducted on a 10.3-In E-Ink Carta module [30], a commonly used component in EPD tablets and e-readers. To minimize interference, updates were applied to a pixel array rather than individual pixels, and the average

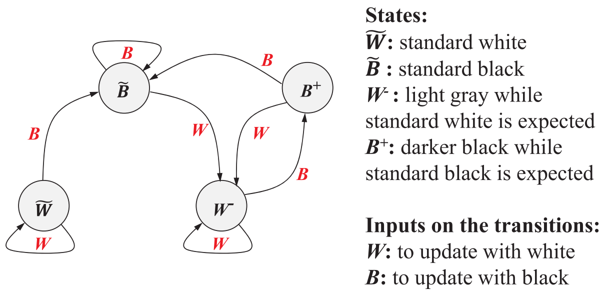


Fig. 7. Automaton for the test-bed EPD device.

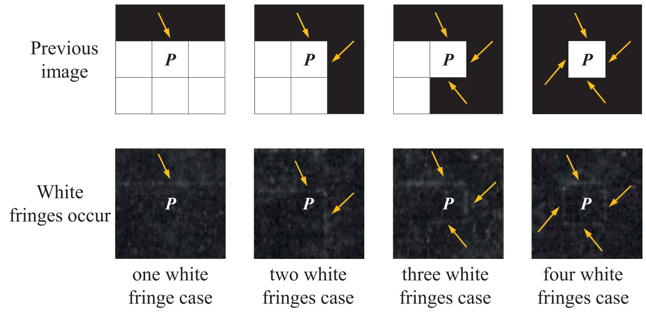


Fig. 8. Illustration of white fringes.

displayed color was recorded after each update step. We record the displayed color in each step of an update history.

Through this analysis, we found specific update patterns on our test EPD device. By classifying all the collected displayed colors, four predominant colors of a pixel are recognized: standard black ($\widetilde{B}$), standard white ($\widetilde{W}$), darker than standard black (B^+), and a light gray close to standard white (W^-).

Then, we use these four colors to represent the display result in each step of an update history. We observed that the resulting color after an update is primarily determined by the color displayed in the preceding frame, a phenomenon we describe as “1-step-history-dependent.” This observation laid the groundwork for constructing an automaton to describe the color transition process for update sequences, as illustrated in Fig. 7. In this automaton, the states represent the four identified colors, and transitions indicate the color changes resulting from updating the pixel to black or white. We express these transitions as $C_i \xrightarrow{B/W} C_j$, where C_i is the initial color, B/W denotes the update command, and C_j is the resultant color.

The automaton unveiled some characteristics of the EPD. For example, a pixel updated from black to white transitions to a light gray state (W^-) and cannot revert to standard white ($\widetilde{W}$), suggesting a disparity in waveform control over the black and white particles. Additionally, a specific sequence ($\widetilde{B} \rightarrow W^- \rightarrow B^+$) indicated that a black–white–black update cycle results in a darker black (B^+) than the standard black, a color observable only under slow refresh conditions. This supports the theory that such an update sequence induces a form of particle activation, enhancing the mobility of black particles.

To validate the automaton’s accuracy, we tested it with update histories of up to eight steps and found no violations. However, longer sequences were not tested due to the exponential increase in possible histories. Within the tested range, the observed behaviors remained consistent with the model.

Given the widespread adoption of the E-Ink Carta module in commercial devices, the behaviors captured by our automaton will be observed in many devices employing the same module type. Nonetheless, other EPDs may exhibit behaviors that depend not only on their most recent update but also on several previous updates. We refer to this characteristic as “N-step-history-dependent.” An automaton that describes this property would need to differentiate update histories that span multiple steps. One potential approach could incorporate the update history information into the automaton’s states. For example, the state would indicate the pixel’s current displayed color and the sequence of updates that lead to this color. This method could noticeably increase the number of states due to the explicit representation of update histories, potentially expanding the automaton to an impractical size for runtime use. Some form of abstraction might be necessary to reduce the size of the automaton while maintaining a manageable loss of precision. Exploring such devices will be our future work.

C. Modeling Fringing

Fringing is a ghosting effect that emerges from the spatial interactions between adjacent pixels. Previous studies [29] have reported the existence of fringing, yet a formal characterization of its impact on display behavior remains unexplored. Our study discloses the conditions that lead to the formation of fringes around pixels and the conditions for their removal. While both white and black fringes can manifest, our discussion will predominantly focus on white fringes for clarity.

To examine the fringing effect, we designed experiments that apply different update patterns surrounding a target pixel. This was achieved by creating images with a chessboard pattern, alternating black and white pixels to simulate different neighbor conditions for the target pixel, as shown in Fig. 8. Initially, this chessboard image is displayed on the EPD, followed by updating the entire screen to black. This sequence results in the appearance of white fringes at the boundaries of certain pixels, demonstrating the fringing effect.

The analysis of the patterns in Fig. 8 reveals that white fringes develop at a pixel’s border exclusively when it transitions from white to black, while its adjacent pixel remains black throughout the process. It is important to note that the formation of white fringes on each of the pixel’s four borders is independent. Our exploration extended to methods of eliminating these white fringes. The findings indicated that simply updating the affected pixel and its neighboring pixels to black does not suffice to erase white fringes. The only successful approach to remove a white fringe involves updating the two pixels aside from the fringe to white, followed by a collective update to black.

Therefore, the emergence and removal of white fringes around a pixel, denoted as p, can be mathematically modeled by (1). Let x represent the four possible directions (top, bottom, left, and right) relative to pixel p, denoted as T, B, L, and R, respectively. The pixel adjacent to p in direction x is represented by p_x. The expressions $C(\text{pre}(p))$ and $C(p)$ denote the colors of pixel p in the preceding and the current frame, respectively. The variable $wf(p, x)$, which transitions from 0 and 1 and from 1 to 0, signifies the formation and elimination of a white fringe along the x direction of pixel p

$$wf(p, x) = \begin{cases} 0 \to 1 & C(\text{pre}(p)) = W \;\;\&\;\; C(p) = B \\ & \&\;\; C(\text{pre}(p_x)) = C(p_x) = B \\ 1 \to 0 & C(\text{pre}(p)) = C(\text{pre}(p_x)) = W \\ & \&\;\; C(p) = C(p_x) = B. \end{cases} \quad (1)$$

Given that black fringes follow the same principles as white fringes, we can model the occurrence and removal of black fringes as follows, where the variable $bf(p, x)$ represents the formation and elimination of black fringes:

$$bf(p, x) = \begin{cases} 0 \to 1 & C(\text{pre}(p)) = B \;\;\&\;\; C(p) = W \\ & \&\;\; C(\text{pre}(p_x)) = C(p_x) = W \\ 1 \to 0 & C(\text{pre}(p)) = C(\text{pre}(p_x)) = B \\ & \&\;\; C(p) = C(p_x) = W. \end{cases} \quad (2)$$

It is important to notice that the proposed data-driven modeling approach is not limited to a single EPD module type. For devices from the same EPD module type, the characteristics and underlying principles remain consistent, making the automata used to describe ghosting effects applicable across all devices of that module type. For devices from different module types, unique automata are generated to accurately represent the specific ghosting effects of each type.

D. Prediction and Color Calibration

Leveraging the models we have constructed, we can predict the ghosting effects that each pixel in the image will experience. This process involves analyzing the dithered images of both the current and the preceding frames as input, allowing us to accurately predict the presence of ghosting effects for every individual pixel (as shown in Fig. 6).

Then, we need to accurately measure the color deviations introduced by these effects. Given that misdriving and fringing impact distinct regions of a pixel, their cumulative influence on color perception must be carefully evaluated. To achieve this, we undertake a comprehensive experimental procedure designed to capture the full spectrum of ghosting effects that may manifest on an EPD screen.

This process involves creating a series of test patterns encompassing all conceivable ghosting scenarios and displaying these patterns on the EPD. Subsequently, we employ a high-precision scanner to capture the display output. The scanned images are then processed to calibrate the grayscale values of the pixels under various ghosting conditions. The actual calibration details and results are reported in Section VI.

The calibrated color data serves a dual purpose. First, it enables us to quantify the specific color deviations associated with each ghosting effect. Second, this information forms the basis for the color adjustments required in the subsequent ghosting effect reduction phase. Also, note that color discrepancies among different devices of the same module type do not alter the fundamental principles of ghosting effect occurrences; hence they do not impact the modeling.

V. Ghosting Effect Reduction

In this section, we present our techniques for reducing the ghosting effect, building upon the results of our ghosting effect modeling. We begin by detailing our primary approach, followed by discussing key challenges encountered and the solutions we have proposed. We will first focus on mitigating the ghosting effect for a single-frame update. Toward the end of this section, we will elaborate on how these techniques are extended and applied across multiple consecutive frames to achieve the comprehensive ghosting effect reduction approach.

A. Primary Approach

Our main strategy for mitigating ghosting effects involves adjusting the colors of pixels in the original image to offset the color deviations caused by ghosting. The general process is shown in Fig. 6. Based on the predicted ghosting effects and the quantified color deviations for each pixel, we modify the color of each pixel in the source image to counterbalance the impact of ghosting. For instance, if ghosting tends to lighten a pixel's color, we darken its grayscale value accordingly. Following these adjustments, the modified image undergoes a standard dithering process to generate a binary (black and white) image required by the fast refresh mode. Our aim is that these color corrections will reduce the visibility of ghosting effects once the updated image is displayed on the EPD.

Nonetheless, some issues still exist, making the primary approach inadequate to mitigate the ghosting effects. The coming two sections will explain these issues and also provide our proposed solutions, which are integrated into the primary approach for a comprehensive solution.

B. Addressing Modification Issues at the Color Extremes

1) Problem: Our exploration into misdriving, represented by our analytical model in Fig. 7, highlights unique challenges on color extremes. We observed that when a pixel's color ends up in W^- (a light gray close to white), attempting to update this pixel to standard white $\widetilde{W}$ is possible. This brings a dilemma: the desired update color is already at its brightest, and there is no further "whiteness" to add to reduce the ghosting effect manifesting as W^-. If widespread across a section of the image, such situations can lead to significant color discrepancies, readily perceptible to viewers.

A parallel issue emerges in the context of fringing. Suppose a standard black pixel exhibits white fringes at its edges; a grayish appearance will be displayed. If the neighboring pixels do not update with the removing pattern specified by the fringing model [referenced as (1)], these white fringes will persist. Note that the pixel is already a standard black pixel. In this case, there is no space for further darkening to mitigate the fringing effect.

These observations necessitate the development of new techniques that can be integrated with our primary approach to effectively address the limitations imposed by color extremes in ghosting effect reduction.

2) Solution—Adjusting Clean Pixels: To address the identified challenges, our proposed solution involves modifying "clean" pixels in regions affected by the aforementioned issues

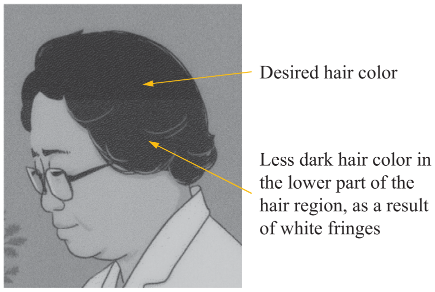


Fig. 9. Explaining the perceivable ghost within a semantic region.

rather than attempting to further adjust the "dirty" pixels already at their color extremes.

Our extensive research on EPD devices showed that ghosting effects are most noticeable to users when they disrupt the uniformity of color in areas where a consistent color is expected. Consider the scenario depicted in Fig. 9, which illustrates a region of human hair composed of black pixels. Due to the fringing effect, the lower part of the hair region may appear grayish, creating a noticeable discrepancy in the overall hair area. In such cases, the ghosting effects become apparent to the viewer. Given that the pixels in this area are already at their darkest, further darkening to counteract the fringing is not feasible. Instead, we lighten the hair's upper part, harmonizing the color across the entire area and reducing the visibility of ghosting.

At first glance, adjusting clean pixels might seem counterintuitive and too aggressive. However, this approach is justified by a couple of practical considerations. First, EPD devices are not known for their precise color reproduction, and even in the absence of ghosting, the colors displayed can deviate from the original image. Hence, minor modifications to the image are unlikely to significantly impact the viewer's interpretation of the screen content. Second, in scenarios where fast refresh modes are in use—often associated with dynamic content like scrolling through web pages—the exact color fidelity of the content is less critical to the user experience. Under these circumstances, slightly modifying the clean pixels to alleviate ghosting becomes a practical and effective solution. The validity of this approach is further demonstrated by the experimental results provided in Section VI, justifying the rationale of adjusting clean pixels.

3) Need for Image Segmentation: The strategy of adjusting clean pixels should not be indiscriminately applied across the entire image, which could lead to potentially disruptive results to image quality. For instance, brightening all pure black pixels to counteract white fringes around some of them results in unnecessary adjustments. We observed that color discrepancies are particularly noticeable within coherent semantic regions, such as the example of the hair region in Fig. 9. Hence, preliminary image segmenting into semantic regions is essential before color adjustments.

An issue in effective image segmentation lies in defining what constitutes "similar" colors. Unlike the uniform hair example, real-world semantic regions often contain a range of colors due to the inherent textures and shading of the objects. Therefore, the segmentation algorithm must be designed to accommodate minor color variations within a semantic region without compromising the region's integrity.

To address this, we adopt a two-phase segmentation strategy. Initially, a seed pixel is chosen randomly, and a group of connected pixels with similar colors is gathered around this seed, like a flooding approach [31]. For practical purposes, we consider pixels within a grayscale value difference of ± 30 to be similar, a threshold determined through empirical testing. This procedure is repeated until the entire image is divided into several preliminary segments, referred to as candidate regions. These candidate regions might represent disjoint parts of the same semantic entity. To unify such regions, the second phase involves calculating the average color for each candidate region and then applying the MeanShift clustering algorithm, with a radius of 16 grayscale values, to merge candidate regions with closely matching average colors. Additionally, the initial segmentation phase can be implemented by initiating the flooding process from multiple seeds simultaneously, allowing for parallel processing and better efficiency.

C. Further Optimization for Quality

An inherent issue with color adjustment strategies is the potential introduction of new ghosts into the image. It has been observed that when dirty pixels—those affected by ghosting—are evenly distributed within a semantic region, the ghosting effect tends to be less noticeable to viewers. Consequently, attempting to counteract these dirty pixels through color adjustments in such regions could inadvertently lead to new, unintended ghosting effects. To reduce this risk, our approach includes assessing whether a semantic region is characterized by a high density of dirty pixels, resulting in nonuniform colors. Regions that do not exhibit significant color nonuniformity are left unaltered to prevent the introduction of additional ghosting artifacts.

Our methodology analyzes the distribution of pixels affected by a predominant ghosting effect within a semantic region. We implemented a sampling-based algorithm that trades some precision for efficiency to accomplish this. Based on the predictions of ghosting effects for each semantic region, we identify the dominant ghosting phenomenon. We then examine smaller subsections of pixels within the region, organized into 6×6-sized grids, to evaluate the prevalence of the primary ghosting effect within these grids. We can gather information on its distribution across the region by calculating the ratio of pixels affected by the dominant ghosting effect within each grid. Following this, we compute the standard deviation of the ratios obtained from all the grids within the semantic region. In this work, a standard deviation below a predefined threshold of 0.15 indicates a relatively uniform distribution of the ghosting effect throughout the region, suggesting that further color adjustments may not be necessary. This approach helps us decide when color corrections would be beneficial, minimizing the risk of introducing new distortions.

D. Handling Consecutive Frame Updates

The previous section discussed our ghosting effect reduction strategy regarding a single frame update. To expand this approach to handle sequences of more than two consecutive

Algorithm 1: Ghosting Effect Reduction

Input: P^i, the i-th/current image to update
Input: P_D^{i-1}, the dithered $(i-1)$-th image updated
Input: $G_{P^{i-2},P^{i-1}}$, the predicted ghost information obtained when updating the $(i-1)$-th image upon the $(i-2)$-th image
Output: $P_D^{i}{}'$, the ghosting-aware dithered image of the i-th frame

// STEP 1: ghosting effect prediction (Sec. IV)
$P_D^{i-1} = \textbf{Dithering}(P^{i-1})$;
$P_D^{i} = \textbf{Dithering}(P^{i})$;
$G_{P^{i-1},P^i} = \textbf{GhostPredict}(P_D^{i-1}, P_D^i, G_{P^{i-2},P^{i-1}})$;

// STEP 2: ghosting effect reduction (Sec. V)
$\{S_1, \ldots, S_n\} = \textbf{ImageSegmentation}(P^i)$; // (Sec. V-B3)
// S_j is the j-th semantic region of P^i
for $j = 1, \ldots, n$ **do**
  // (Sec. V-C)
  if ***NeedToAdjust***(S_j, G_{P^{i-1},P^i}) **then**
    flag $= \textbf{CleanOrDirty}(S_j, G_{P^{i-1},P^i})$; // (Sec. V-B)
    $S_j' = \textbf{Adjust}(\text{flag}, S_j, G_{P^{i-2},P^{i-1}}, G_{P^{i-1},P^i})$;
    // Performing color adjustment to the original image (Sec. V-A, V-B, V-C)

// STEP 3: generating output
$P^{i\prime} = \{S_1', \ldots, S_n'\}$;
$P_D^{i}{}' = \textbf{Dithering}(P^{i\prime})$;
return $P_D^{i}{}'$;

frame updates, a crucial aspect is ensuring the accurate carryover of ghosting information from one frame to the next. Consider a scenario involving a pixel p across three sequentially updated frames: A, B, and C. Ideally, p is meant to transition from black in frame A to white in frames B and C. Due to misdriving, however, p exhibits a W^- coloration after the update from A to B. To precisely determine p's color after the update to frame C, it is imperative to account for the fact that p was displaying W^- after the B update. Thus, when modeling the ghosting effect for the upcoming update to frame C, the analysis must include the ghosting information obtained from updating from C and the ghosting information obtained when updating frame B.

In the "Ghosting Effect Reduction" phase of our approach, as outlined in the general workflow depicted in Fig. 6, the input for ghosting information encompasses the analysis results of ghosting effects predicted for both the current and the preceding frames. For simplicity, we do not add a new figure to show the adjustment to the workflow illustrated in Fig. 6.

E. Putting All Together

Integrating the components above, we present a comprehensive algorithm for predicting and reducing ghosting effects, detailed in Algorithm 1. To ascertain the ghosting effects for the current frame, referred to as P^i, it is essential to have access to the immediately preceding frame, P^{i-1}, along with the ghosting effect predictions from the prior cycle, $G_{P^{i-2},P^{i-1}}$. Subsequent to the ghosting effect prediction for P^i, color adjustments are made, taking into account the ghosting information $G_{P^{i-2},P^{i-1}}$, especially crucial for spanning multiple consecutive frames. The conversion of images from a 256-color grayscale to a binary format is achieved through a dithering process, employing the widely adopted Floyd–Steinberg algorithm (Dithering function) [27] in image processing.

VI. Implementation and Evaluation

A. Experimental Settings

In the evaluation, we utilized a 10.3-In Waveshare EPD equipped with the E Ink Carta module [30], with a 1872×1404 pixels resolution. Given its prevalence in a range of EPD-based commercial devices, this module is an apt model for our testing. The EPD's capability to support fast refresh rates makes it particularly suitable for scenarios requiring dynamic screen updates. The device is operated via a Raspberry Pi, which acts as the embedded controller for screen updates.[2]

Our test suite comprises images from common use cases, including Web browsing, blog reading, and image galleries. We observe ghosting effects by sequentially updating images on the EPD. We use an EPSON V19 scanner set to "continuous automatic exposure" mode to capture the display outcomes accurately. This mode automatically determines the optimal exposure parameters for each scan and allows the parameters to be locked for future scans if a "preview" operation is conducted initially. We conducted this preview to ensure uniform scan settings throughout our experiments. For EPD feature exploration and color calibration, we use the 2400 dpi setting, where one EPD pixel is represented by about 140 pixels in the scanned image. For our empirical studies, we used 600 dpi to enhance efficiency, with 9 pixels representing an EPD pixel in the scanned image. Both resolutions are sufficient to clearly capture the ghosting effects.

B. Efficient Application of Display Histories

To effectively model the ghosting effect, it is crucial to investigate the color outcomes associated with various display histories, beginning from an initial state. Given that the EPD screen can be updated to either black or white with each step, there are 2^N potential update sequences for N steps. We adopt a parallelization strategy to gather data on these display histories efficiently using a single EPD test device.

For instance, when examining update sequences encompassing up to seven steps, we divide the EPD screen into 128 equally sized rectangular sections, each containing 117×175 pixels. These sections are assigned IDs ranging from 0 to 127, with each ID expressible as a 7-digit binary number. Each binary digit within an ID corresponds to a specific update step, where "0" signifies an update to black, and "1" represents an update to white. Thus, the 7-digit binary ID of a section distinctly delineates a unique update sequence over

[2] Direct implementation on commercial EPD devices, such as tablets or phones, was not feasible due to proprietary restrictions on the mobile operating systems and EPD drivers.

seven steps. To simulate all possible update histories of up to seven steps with minimal images, we create seven chessboard-patterned images. Each image, representing a particular update step, determines the target color for each section based on the corresponding digit in the section's ID. This allows for generating all 7-step histories using merely seven images on a single EPD screen.

For the initial image, the EPD's update requires two color outcomes for all sections—black or white. Subsequent updates incrementally double the number of possible update histories. Following each update, we scan the EPD's display to document the colors manifested by the latest update. This approach facilitates the efficient collection of displayed colors for all possible 7-step histories. It is important to note the necessity of balancing the number of screen divisions to avoid excessively small sections, which could lead to inaccuracies in the color measurement over scanned images.

C. Color Calibration and Its Results

For effective modeling and reduction of ghosting effects, precisely identifying the actual grayscale value of pixels on the screen is fundamental. To this end, we undertake empirical color calibration, drawing grayscale values directly from high-resolution scans of the display. Although the scanning process may introduce slight brightness variations, these do not detract from our analysis, which focused on relative color differences. The critical aspect is the scanner's fidelity in capturing these differences, ensuring the utility of the scanned images for our evaluations. To enhance the reliability of our calibration, we repeat the procedure ten times for each scenario, averaging the results to derive stable grayscale values.

To calibrate the colors for deeper blacks and lighter grays, we create extensive pixel areas manifesting these specific colors and compute the average grayscale value across all pixels. Fringes, which primarily affect pixel boundaries without extending into the pixel's core, present a unique challenge due to the tiny size of individual pixels. Instead of isolating these narrow borders for separate color modeling, we calibrate a pixel color by accounting for varying fringe counts. This is accomplished by generating expansive chessboard-patterned areas designed to induce 1–4 fringes per pixel. After scanning these areas, we average the grayscale values of the targeted pixels to ascertain the resultant colors. These values are denoted as $gs(\widetilde{B}_{wf_x})$ and $gs(\widetilde{W}_{bf_x})$ for black and white pixels bordered by x number of white and black fringes, respectively. Furthermore, the interplay between misdriving and fringing necessitates a calibration process that covers all conceivable combinations. We denote the grayscale values for a W^- pixel affected by both x black fringes and misdriving as $gs(W^-_{bf_x})$, and similarly, $gs(B^+_{wf_x})$ for a B^+ pixel experiencing both x white fringes and misdriving.

The detailed outcomes of our color calibration are presented in Table I. It is worth noting that slight variations in color may exist among pixels of the same category. To counteract this variability, we intentionally create large pixel clusters for calibration and calculate the average grayscale values for the entire group of pixels under examination whenever necessary, ensuring a more accurate calibration of the colors with complex ghosting situations.

TABLE I
COLOR CALIBRATION RESULTS

Color	Description	Grayscale
B	standard black color in the color space	0
W	standard white color in the color space	255
$\widetilde{B}$	"standard" black displayed	36
$\widetilde{W}$	"standard" white displayed	178
B^+	A darker black resulted by misdriving	35
W^-	A greyish white resulted by misdriving	172
$\widetilde{B}_{wf\_i}$	$\widetilde{B}$ pixel with i white fringes, $i = 1, 2, 3, 4$	46, 47, 49, 50
$B^+_{wf\_i}$	B^+ pixel with i white fringes, $i = 1, 2, 3, 4$	45, 46, 47, 49
$\widetilde{W}_{bf\_i}$	$\widetilde{W}$ pixel with i black fringes, $i = 1, 2, 3, 4$	168, 165, 161, 158
$W^-_{bf\_i}$	W^- pixel with i black fringes, $i = 1, 2, 3, 4$	161, 156, 152, 149

Recalibration: It is important to note that color calibration is specific to each EPD device. Recalibration is needed when there is a noticeable color deviation between a newly manufactured EPD device and a previously calibrated one. This ensures accurate color compensation during ghosting effect reduction. In industrial settings, devices are often produced in batches. Significant color deviations within the same batch are rare for mature production lines. Consequently, it is common practice to calibrate once per batch prior to product delivery. For devices from different EPD modules that may vary in their working principles, recalibration is typically required. For applications requiring optimal ghosting effect reduction for each specific device, the ideal practice would be to calibrate each device independently, similar to the individual predelivery calibration for LCD monitors in specialized professional domains, ensuring optimal color accuracy.

D. Evaluating Ghosting Effect Reduction

1) Evaluation Metric: The visibility of ghosting effects to users is primarily from the occurrence of color discrepancies within specific regions, deviating from expected uniformity. A crucial component of our evaluation is adopting a metric that quantitatively assesses the efficacy of our reduction techniques.

To achieve this, we utilize the "deep image structure and texture similarity (DISTS)" index [32], [33], a metric designed to evaluate image similarity. This index, based on the methodology described in [32] and [33], is adept at capturing similarities between images. When comparing two images, A and B, the index produces a non-negative value indicating their similarity. A smaller index value suggests that image B is more similar to image A.

In the context of assessing ghosting effects, a more pronounced ghosting anomaly would lead to a significant divergence between the displayed image and the original, thereby resulting in a higher DISTS index value (denoted as v_g). On the other hand, successful mitigation of ghosting effects should bring the displayed image closer to resembling the original, manifesting as a reduced DISTS index value (denoted as v_r). The relative decrease of v_r compared to v_g serves as an indicator of the ghosting reduction's effectiveness. The DISTS index is generally reliable for evaluating small to medium-sized images, and its precision may diminish for higher-resolution images due to localized textural variations.

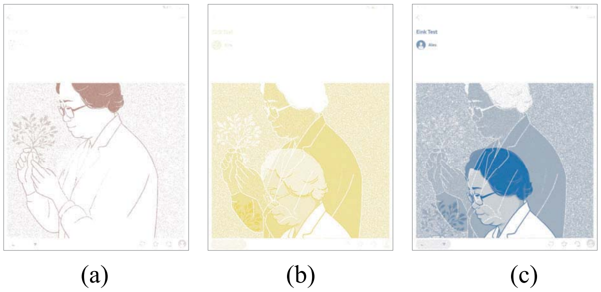


Fig. 10. Visualizing different ghosting effects. (The colors in the figure are not from the original image; we use different colors to represent the occurrence of different ghosting effects on the pixels.) (a) Fringing white fringe. (b) Fringing black fringe. (c) Misdriving deviated white.

To circumvent this, we downscale the scanned images to a resolution of 200 × 150 pixels prior to calculating the DISTS index values, ensuring the metric's accuracy and relevance to our analysis.

2) Case Study: To show the effectiveness of ghosting effect reduction, we present a case study demonstrating a web page scrolling scenario, where the page transition displays image A followed by image B, as depicted in Fig. 11.

Initially, we analyze the ghosting effects through dot cloud visualizations (Fig. 10), where (a)–(c) illustrate pixels affected by white fringes, black fringes, and misdriving, respectively. The analysis reveals a significant presence of ghosting effects.

The impact of ghosting is evident in image B_S (Fig. 11), where the figure's hair, expected to be close to black, appears grayish in the lower portion due to prevalent white fringing effects. This discrepancy highlights the visual significance of ghosting effects. Similarly, the clothing area, intended to be standard white, is predominantly displayed in light gray due to misdriving, with only a small section maintaining the desired standard white appearance. The contrast within these regions manifests visible ghosts to viewers. Image B' represents the outcome of applying our ghosting effect reduction technique, with adjustments made to the hair and clothing regions for color correction. Notably, the clothes region required modifying clean pixels to diminish the ghosting effect. The comparison between B' and B shows the adjustments made, and B'_S shows a reduction in the perceptible ghosts. Ghosting observed in background areas has uniform distribution and thus is less noticeable. Our methodology, as outlined in Section V-C, effectively identifies and excludes such evenly affected regions during the color adjustment phase.

Further examples from diverse scenarios are shown in Fig. 12, which includes three cases (arranged in rows). The first column presents the ground truth (the image to be displayed), the second column shows the displayed result without our intervention (highlighting visible ghosts), and the third column displays the result after applying our ghosting effect reduction approach, showing a decrease in visible ghosting effects. Due to space constraints, the previous frame causing the ghosting in the current frame is not shown. Accompanying DISTS index values are provided for quantitative evaluation.

Several insights can be obtained from these results. First, addressing ghosting effects in images characterized

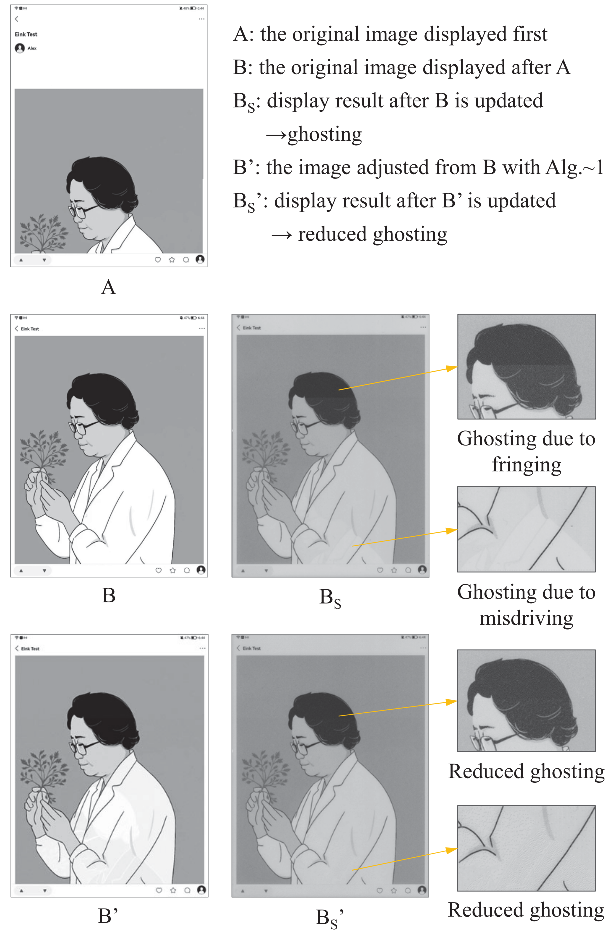
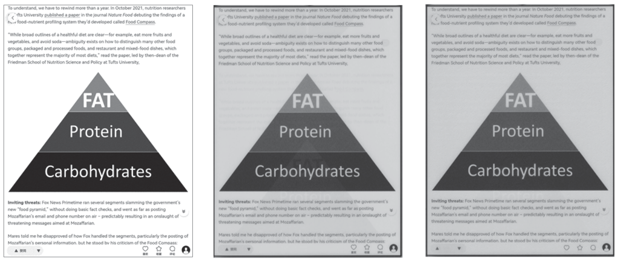


Fig. 11. Use case of ghosting effect reduction.

by complex textures is more challenging, attributed to the intricacies involved in image segmentation. Second, within the typical framework of web page images, which often comprise diverse regions with either textual content or graphical elements, the application of semantic segmentation plays an important role. By distinguishing the ghosting phenomena in these regions, it becomes possible to apply different adjustment strategies, thereby enhancing the efficacy of the ghosting effect reduction.

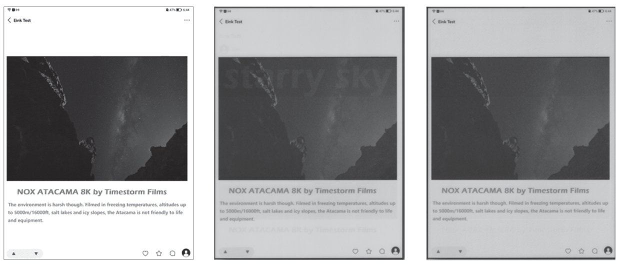
3) Empirical Study: In order to investigate the prevalence of ghosting effects across a wide variety of images and to evaluate the efficacy of our proposed solutions, we undertook an empirical study with 100 pairs of images. Twenty pairs of images are obtained from web pages, photographs, and cartoon pictures representing various application scenarios. Eighty pairs are synthetic images produced by randomly putting around 100 geometric objects (circles, triangles, rectangles, and polygons) in an image, with each object randomly assigned grayscale values ranging from 0 to 255 in order to generate diverse complexity of the image content.

Each pair of images was sequentially displayed on the EPD screen, and the DISTS index was calculated to quantify the ghosting effect with and without our ghosting effect reduction technique. Before applying our method, the average DISTS index across all image pairs is 0.096. Our approach decreased



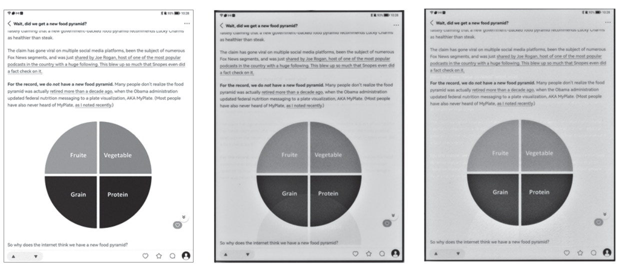
DISTS = 0.054 DISTS = 0.033

The page moves up, the triangle has ghosting effect in its original position, and text also leaves ghosting effect on the background.



DISTS = 0.080 DISTS = 0.042

In the fist image, the words “starry sky” are shown in the upper part of the screen. After the second image is applied, the ghost of the words remains.



DISTS = 0.051 DISTS = 0.049

The pie figure scrolls up with the page, leaving ghost at its previous location. Ghosting effect from the text also leaves in the background.

Fig. 12. More cases for ghosting effect reduction.

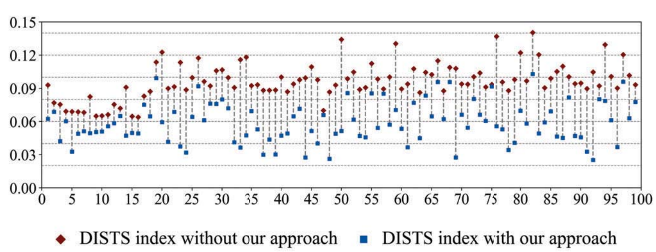


Fig. 13. Evaluation results of the empirical study.

the average DISTS index to 0.058, showing our method's capability to reduce the ghosting effects. All DISTS index value pairs are visualized in Fig. 13. The 100 pairs of images and the scanned display results are provided via a weblink.[3]

E. Execution Performance Evaluation

To evaluate the operational efficiency of our ghosting effect reduction method and its impact on the display update process, we carried out performance evaluation a Huawei Mate 40 Pro

[3]https://anonymous.4open.science/r/ghostbuster-B3D9

TABLE II
PERFORMANCE EVALUATION RESULTS

Components	C1	C2	C3	C4	C5	Total
Delays (ms)	14.7	9.0	77.7	10.5	14.4	126.3

smartphone, featured by an octa-core Kirin 9000 CPU and an ARM Mali-G78 MP16 GPU. The delay introduced by our approach comprises several steps.

1) *C1:* Dithering the original image.
2) *C2:* Conducting an analysis of ghosting effects.
3) *C3:* Segmenting the image.
4) *C4:* Modifying the original image based on the analysis.
5) *C5:* Dithering the modified image for display.

To enhance the performance of our approach, we first downscaled the original image prior to the segmentation process, which significantly boosts performance with a minimal compromise in accuracy. Second, given the data-parallel nature of all five components of our algorithm, they are well suited for GPU acceleration. Considering the widespread availability of mobile GPUs in contemporary devices, such as smartphones, tablets, and laptops, leveraging GPU acceleration is a standard and practical choice. We implemented these steps using the OpenCL 3.0 library.

An extensive analysis was conducted to measure the time delays for each component across our 100-case empirical study. The average delays recorded for components C1–C5 on the test device are given in Table II. The average total time delay is 126.3 ms. In comparison, the standard refresh time for an EPD device, particularly in fast refresh mode, falls between 120 and 150 ms, as dictated by the waveform application process. To reduce the impact of the additional delays from ghosting effect reduction, we propose a pipeline strategy that synchronizes with the screen refresh cycle. Specifically, while the EPD controller is busy updating the current frame, the UI system can simultaneously prepare and render the subsequent frame (with ghosting effect reduction) in the background. This method requires slight modifications to the UI system architecture, such as in Android devices, to enable parallel processing of screen refreshes and frame preparation using dual buffers. This setup ensures that the refresh rate, even in fast refresh mode, remains consistent, unaffected by the integration of ghosting effect reduction measures.

VII. RELATED WORK

EPDs have dominated the eReader market, for their paper-like display qualities and energy efficiency [1]. Initially, the focus of hardware advancements in EPDs was on improving the resolution and contrast. However, the inherent limitations related to slow refresh rates and noticeable screen flickering, primarily due to the nature of driving waveforms, have hindered their application range. While these limitations pose minor concerns for eReaders requiring infrequent screen updates, they become significant in applications demanding quick refresh, as outlined in Section I.

In response to these challenges, substantial research has been directed toward the development of short waveforms for EPD operation [28], [34], [35], [36], [37], with the goal of reducing waveform duration. Some efforts focus on shortening the activation phase, while others propose eliminating it entirely to achieve quicker refresh times and eliminate flicker.

Despite their advantages, short waveforms compromise precise control over EPD particles, leading to ghost images. Recent studies explored waveform enhancements to reduce ghosting effects. For example, research by Yang et al. [38] investigated how different waveform parameters influence ghosting. However, their solutions reintroduced flicker due to the inclusion of a short shaking phase. Shen et al. [29] tackled fringing by fine-tuning waveforms to account for the electric field dynamics. Given that waveform adjustment often involves manual tuning, Cao et al. [25] leveraged CNN to automate the identification of ghosting effects from extensive display outcomes for systematic waveform optimization.

While existing research predominantly centers on refining waveforms to address ghosting, commercial EPD solutions frequently offer users multiple refresh modes to suit different usage scenarios. Nevertheless, these measures typically fall short of completely eradicating ghosting due to the intrinsic limitations of short waveforms. Our approach diverges from traditional methods by acknowledging the inevitability of ghosting with short waveforms. Instead, we employ software-based methods to counteract the effects of ghosting, aiming for a display outcome with less noticeable ghosts, thereby enhancing the overall viewing experience.

VIII. Conclusion

EPDs are becoming popular for a wide array of embedded devices. The necessity for fast refresh rates to accommodate dynamic content has introduced the challenge of ghosting effects, which can significantly degrade the user experience. To address this issue, we presented a software-centric methodology that not only precisely models the ghosting phenomenon but also effectively diminishes its impact by modifying the original image. The efficacy of our approach has been validated on actual EPD devices. Our solution provides a promising avenue for ensuring that users can fully enjoy the inherent advantages of the EPD technology with reduced disturbance from the ghosting effect problems.

References

[1] B. Yang, *E-Paper Displays*. Hoboken, NJ, USA: Wiley, 2022.

[2] Z. Zhao, Y. Zhou, G. Tan, and J. Li, "Research progress about the effect and prevention of blue light on eyes," *Int. J. Ophthalmol.*, vol. 11, no. 12, p. 1999, 2018.

[3] N. Wong and H. Bahmani, "A review of the current state of research on artificial blue light safety as it applies to digital devices," *Heliyon*, vol. 8, no. 8, 2022, Art. no. e10282.

[4] A. Cougnard-Gregoire et al., "Blue light exposure: Ocular hazards and prevention—A narrative review," *Ophthalmol. Ther.*, vol. 12, no. 2, pp. 755–788, 2023.

[5] "E-reader market size and forecast." 2024. [Online]. Available: https://www.verifiedmarketresearch.com/product/e-reader-market/

[6] "Berny E-lnk business watch product page." Berny. 2024. [Online]. Available: https://www.bernywatch.com/products/berny-men-e-lnk-square-business-watch-e002

[7] "Sony FES watch product page," Sony. 2024. [Online]. Available: https://www.sony.com.hk/en/electronics/support/other-products-xperia-smart-devices/fes-wa1s/specifications

[8] "Epson smart canvas product page." Epson. 2024. [Online]. Available: https://w3.epson.com.tw/smartcanvas

[9] "Fossil FTW7014 product page." Fossil. 2024. [Online]. Available: https://www.fossil.com/en-us/products/hybrid-smartwatch-hr-charter-rose-gold-tone-stainless-steel-mesh/FTW7014.html

[10] "Hisense A5 product page." Hisense. 2024. [Online]. Available: https://mall.hisense.com/items/4582

[11] "Hisense A7 product page." Hisense. 2024. [Online]. Available: https://mall.hisense.com/items/4030

[12] "Hisense A9 product page." Hisense. 2024. [Online]. Available: https://mall.hisense.com/items/4770

[13] "Huawei MatePad paper product page." Huawei. 2024. [Online]. Available: https://consumer.huawei.com/en/tablets/matepad-paper

[14] "Lenovo YOGA paper product page." Lenovo. 2024. [Online]. Available: https://item.lenovo.com.cn/product/1027315.html

[15] "BOOX tab X product page." BOOX. 2024. [Online]. Available: https://www.boox.com.hk/products/eink-android-tablet-tab-x

[16] "iFLYTEK X2 product page." iFLYTEK. 2024. [Online]. Available: http://www.iflyink.com/#/product_detail/x2

[17] "DASUNG A4 product page." Dasung. 2024. [Online]. Available: http://www.dasung.com/h-pd-48.html#_jcp=2

[18] "Lenovo ThinkBook plus gen 2 product page." Lenovo. 2024. [Online]. Available: https://www.lenovo.com/hk/en/laptops/thinkbook/thinkbook-series/ThinkBook-Plus-Gen-2/p/XXTBXPLI300

[19] "Lenovo yoga book C930 product page." Lenovo. 2024. [Online]. Available: https://www.lenovo.com/hk/en/tablets/lenovo-tablets/yoga-tablets-series/Yoga-Book-C930/p/ZZIWZWBYB1J

[20] "MODOS product page." MODOS. 2024. [Online]. Available: https://www.modos.tech

[21] "Dasung Paperlike 253 E-ink monitor." 2024. [Online]. Available: https://shop.dasung.com/pages/more-about-paperlike-253

[22] "Dasung launches paperlike U—The world's first monitor with 25.3-inch curved E ink display." 2024. [Online]. Available: https://goodereader.com/blog/technology/dasung-launches-paperlike-u-the-worlds-first-monitor-with-25-3-inch-curved-e-ink-display

[23] "BOOX MiraPro 25.3-inch E-ink monitor." 2024. [Online]. Available: https://zh.boox.com/mirapro

[24] B. Yang et al., "Understanding the mechanisms of E-ink operation," in *Proc. Int. Disp. Workshops (Web)*, 2019, pp. 1–3.

[25] J. Cao et al., "A convolutional neural network for ghost image recognition and waveform design of electrophoretic displays," *IEEE Trans. Consum. Electron.*, vol. 66, no. 4, pp. 356–365, Nov. 2020.

[26] W. Kao and J. Tsai, "Driving method of three-particle electrophoretic displays," *IEEE Trans. Electron Devices*, vol. 65, no. 3, pp. 1023–1028, Mar. 2018.

[27] R. Floyd, "An adaptive algorithm for spatial greyscale," in *Proc. SID*, 1975, pp. 75–77.

[28] W. Kao, C. Liu, S. Liou, J. Tsai, and G. Hou, "Towards video display on electronic papers," *J. Disp. Technol.*, vol. 12, no. 2, pp. 129–135, 2016.

[29] S. Shen et al., "Improving electrophoretic particle motion control in electrophoretic displays by eliminating the fringing effect via driving waveform design," *Micromachines*, vol. 9, no. 4, p. 143, 2018.

[30] "The Carta product page." E-Ink. 2024. [Online]. Available: https://www.eink.com/brand?bookmark=Carta

[31] F. Meyer, "Color image segmentation," in *Proc. Int. Conf. Image Process. Appl.*, 1992, pp. 303–306.

[32] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, May 2022.

[33] dingkeyan93. "Deep image structure and texture similarity (DISTS) metric." Mar. 2024. [Online]. Available: https://github.com/dingkeyan93/DISTS

[34] W. Kao and C. Liu, "Driving waveform design for playing animations on electronic papers," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, 2015, pp. 599–600.

[35] L. Wang, Z. Yi, M. Jin, L. Shui, and G. Zhou, "Improvement of video playback performance of electrophoretic displays by optimized waveforms with shortened refresh time," *Displays*, vol. 49, pp. 95–100, Sep. 2017.

[36] W. He et al., "Driving waveform design of electrophoretic display based on optimized particle activation for a rapid response speed," *Micromachines*, vol. 11, no. 5, p. 498, 2020.

[37] W. Zeng et al., "Design of driving waveform for shortening red particles response time in three-color electrophoretic displays," *Micromachines*, vol. 12, no. 5, p. 578, 2021.

[38] S. Yang et al., "P-83: Ghosting reduction driving method in electrophoretic displays," in *SID Symp. Tech. Dig.*, 2012, pp. 1361–1364.