A Saliency-Driven LCD Power Management System

Yang Xiao, Siddharth Advani, Donghwa Shin, Naehyuck Chang, Fellow, IEEE, John (Jack) Sampson, and Vijaykrishnan Narayanan, Fellow, IEEE

Abstract—Large liquid crystal display (LCD) technology is being widely used in every corner of our modern life, ranging from personal laptops to flat-panel televisions. Among all the components in an LCD display system, the backlight panel is the dominant power consumer, irrespective of lighting technology or class. In this paper, a saliency-based field-programmable gate array accelerator for revolutionary LCD power management is proposed that allows dynamic modulation of the different zones of the backlight panel. This hardware accelerator-based system is capable of processing a high-definition video stream in real-time and uses less than 50% of the power that a normal LCD display system consumes, with minimum overhead. We also compare our proposed approach with other state-of-the-art power-aware methods and show numerous advantages using our data-driven strategy.

Index Terms—Field-programmable gate array (FPGA), liquid crystal display (LCD), power-aware systems, saliency.

I. INTRODUCTION

SINCE the invention of the first television, the display system has manifested into a critical human–computer interface capable of bringing vivid and abundant visual information to users. As technology evolved, flat-panel liquid crystal display (LCD) display systems were invented and are now widely used in various multimedia systems, ranging from large screen home theater systems to personal laptops and mobile devices. As projected in [1], LCD TVs are becoming progressively bigger in size every year, with customers demanding powerful visual experiences as video technology evolves. As the smartphone industry explores newer dimensions of innovation, there is an increasing demand for larger screens in this multimedia category too [2].

The LCD panel in a common LCD display system will not emit light by itself. Therefore, a backlight panel and a light-diffusing component are used under the LCD panel as a source of lighting in the system. Currently, there are two kinds of lighting sources employed for the backlight panel. An older method uses cold cathode fluorescent lamps (CCFLs); while a more recent one utilizes light-emitting diodes (LEDs) since LEDs can offer greater dynamic contrast, wider color gamut and less power consumption. Although this new backlight technology can provide better power efficiency, it needs to be pointed out that the backlight panel still remains the largest portion of the entire system power consumption [3]. With the increasing demand for larger screens, optimizing and enhancing the power efficiency of the LCD display system, especially televisions [4], is thus attracting continuing research efforts.

While approximate computing is becoming a powerful paradigm to save energy in the vision space [5], another promising way to solve the power problem specific to LCD panels is to dim the LED backlight. Many different methods have been proposed in the same vein. Active dimming approaches adjust the luminance level of the backlight based on pixel information such as contrast, color, or brightness. Passive dimming methods modify the luminance level by monitoring user attention with a camera or sensors. The backlight is dimmed when users are away and restored to full level when users are in front of the display.

In [6], an active dimming strategy is proposed, where visual saliency is used to adaptively change the luminance level of the backlight panel at a zone granularity. Most objects of interest have distinct features that stand out in contrast to their background. Xiao et al. [6] used three feature channels—intensity, color, and orientation—for locating salient or interesting objects/regions in the scene. These low-level features, when applied to images, perform extremely well, as was demonstrated in [6]. However, due to the static nature of these channels, the system, when operating on a video stream, is constrained in that it fails to behave like a person who can focus attention on new objects entering the frame, or moving objects across a series of frames. To overcome these handicaps, we extend [6] with the following contributions.

1) LCD-based devices in these days invariably have streaming video applications running on them and consume more power than when a static image is being observed. As highlighted in [7], two additional processing channels, motion and flicker are effective in finding salient regions in a temporal environment. We thus incorporate these two channels into [6] to account for dynamic occurrences across a stream of video frames.

2) We then highlight the impact of these changes on user experience. Since the original system settings in [6]...
introduce a shimer (discussed later) in the final compensated video, new dimming and compensation coefficients are used. A movement constraint for salient zones between two continuous frames is also adopted for preserving video quality. Using these new compensation coefficients, verification of our system is undertaken with 29 different videos as part of a user perception evaluation test.

3) To address system constraints such as power, performance, and resource tradeoffs, this paper adopts a generic field-programmable gate array (FPGA) channel architecture. On average, a 50% power saving can be achieved with our extended system while maintaining real-time constraints.

The organization of the rest of this paper is as follows. Section II highlights related work on bioinspired systems in general and various LED power saving schemes in specific. Section III introduces saliency, the adopted biological attention model, in detail; the LED backlight system and its power model is outlined in Section IV; Section V shows the design and implementation details of the proposed power management system; all the experimental results are provided in Section VI; finally we conclude the paper in Section VII.

II. RELATED WORK

Developing embedded systems using neuromorphic computing has allowed the advent of a stream of energy-efficient designs. Neuromorphic computing refers to systems that, unlike the von Neumann model, mimic the architecture of biological neural networks (e.g., human brains) for computation rather than implementing a sequence of instructions. In spiking neural networks, information is transmitted via spikes, which combine with per-neuron weights to generate spikes in subsequent layers. These spiking networks are known to provide energy-efficient encodings [8]. Nonspiking networks still benefit from massive inherent parallelism. Examples of neuromorphic systems in the visual space include the following. Zappi et al. [9] presented a gesture recognition system that minimizes power for a wearable runtime application environment. Al Maashri et al. [10] mapped HMAX, a biological model for object recognition, to an FPGA-based platform and show significant power benefits when compared to CMP and GPU platforms. Kyrkou et al. [11] demonstrated an architecture for object detection using depth and edge information, thus providing a power-efficient platform for real-time computing. Very recently, Neil and Liu [12] proposed a spiking deep network that can be targeted for low-power embedded robotic applications.

With regard to LCD televisions, several works in the embedded system field have been presented to save the power consumption of the backlight panel via active dimming [13]–[18]. Dynamic luminance scaling is demonstrated in [13], and the proposed system can achieve 20% to 80% power saving by modulating the luminance of CCFL backlight based on image content. Cho et al. [14] provided a zone dimming methodology on 2-D-LED LCD display. The 2-D array LED backlight panel is divided into X–Y grids. The luminance level of each zone is determined by picking the smaller luminance value between the row controller and the column controller in that zone. The system can achieve 30%–80% power saving. The idea of dimming backlight panel zone by zone is also presented in [15] and [16]. Lee et al. [15] chose the ratio between the average pixel value within a zone and the maximum pixel value representable (255 in 8-bit representation) for the zone’s luminance level. Anggorosesar and Kim [16] took a step further and proposed an active dimming strategy that uses object recognition. It is important to note that after dimming, image compensation is usually applied either at a zone level or a pixel level granularity to preserve the image quality. Gatti et al. [17] proposed to compensate for the reduced luminance by changing the transmittance. Inspired with the idea of pixel compensation by adjusting RGB values of each pixel of the image in [18], this paper also adopts pixel compensation to maintain image quality.

In addition to active methods, passive dimming technologies have also been developed for reducing LCD power consumption. Sensors are used in [19] to monitor user behavior for energy management on personal laptops. While in [20], a camera is adopted in a similar fashion for controlling pixel luminance. With low-power sensors or monitoring devices, these approaches can achieve high power savings when the user is away from the screen. However, the power saved when the user is active is negligible, and tracking actions or eye gazes of multiple users introduces significant complexity.

A neuromorphic method that used fundamental vision features for LCD power consumption was first introduced in [6]. Here, a neurovisual algorithm to model attention was integrated into the LCD system evaluated. The proposed system not only provided stable power savings on different types of high-definition (HD) images but also continued to offer high visual quality, as measured by user experience studies. This paper focuses on extending the work in [6]. Although the same algorithm, dimming strategy, and image compensation techniques are adopted here, the major differences and contributions of this paper are as follows.

1) Extending the original system from three channels to five channels to enable power savings in a streaming video system. The original system fails to focus attention on new objects or moving objects that can occur in such a dynamic application. Introducing two additional channels flicker and motion enables the system to account for differences between two consecutive frames as well as a stream of frames. This allows new objects or moving objects to be located on the saliency map, thus providing accurate dimming zones.

2) A generic FPGA channel architecture is adopted and modified based on the pipeline proposed in [21], parameters are optimized in our system to enable streaming visual information without affecting visual experience.

3) Both image and video quality verification are done on the system. Due to a shimmer effect seen in the final compensated video, different parameters are set compared with the image-based approach. With these new parameters, the shimmer effect is fixed and the
system provides significant power savings while preserving good user experience.

III. SALIENCY

Visual attention, in general, explains how people efficiently perceive the environment and how they shift their focus of attention to different objects when exposed to a complex visual scene. Since the fovea is small, humans do not perceive every object in a scene equally, but instead they prioritize, and their eye fixations will attend to every potential object in the scene. Deducing the exact fixation path is a complex problem since it not only involves figuring out which features are visually important in the scene, but also what the viewer is thinking during the visual observation. Bottom–up saliency is one way of attributing attentional response to different visual stimuli while being agnostic of top–down or task-driven attention.

In the context of real-time embedded vision, saliency can also help reduce the overall computational burden further down the processing pipeline. For example, once the salient regions/objects are identified in a particular video frame, complex visual tasks such as object recognition or action recognition only need to be evoked on these regions rather than the entire frame. This method can not only save a lot of computing resources, but also help achieve real-time performance.

Many saliency models have been proposed over the past few years, and they have been benchmarked using standard metrics [22]. However, choosing the right saliency model for the task at hand is imperative. The visual attention algorithm adopted in this paper stems from the work of [7], [23] and is shown in Fig. 1. We choose this model for the following two reasons.

1) The model is designed to mimic the ability of primates of focusing visual attention to important areas in the field-of-view when exposed to a scene. In addition to three static channels of the bottom–up saliency model, intensity, orientation, and color, two more channels, flicker and motion channels are introduced here.

These two additional channels are modeled to cater to dynamic features in streaming video, as shown in Fig. 2.

2) While this model is computationally intensive, it is highly parallelizable and a custom accelerator can help achieve real-time performance imperative for our application.

The model is composed of five feature channels—intensity, color, orientation, flicker, and motion. To begin with, the original image (in RGB format) is converted to one luminance component (I) and two chrominance components (RG and BY) using the mappings given in [25]. The luminance component is used in the intensity, orientation, flicker, and motion channels while the chrominance components are used in the color channel. The color channel is further subdivided into two channels, while the orientation and motion channels are each subdivided into four channels. Each of these 12 channels then passes through four stages of computation described in detail below.

A. Filtering

In the first step, a pyramid of alternating linear filters and subsampling filters is used across all channels. A Gaussian kernel of of 5 × 5 support is used for all channels except the orientation channel, which uses a Laplacian kernel of the same size. The decimation factor is set to two for the subsampling filters. As a result, within each channel, an image pyramid having nine image scales is formed, where every image scale is half the size of the previous image scale. The flicker channel in this stage goes through a preprocessing step, which computes the absolute luminance difference between previous and current image frames, thus yielding an F component. The orientation and motion channels need a postprocessing step to extract the features of interest (edge orientation/motion direction) from the image pyramid, thus yielding O and M components, respectively.

B. Center-Surround Difference

In the second step, center-surround differencing (CSD) operation and normalization is applied to the pyramids in each channel. In the differencing operation, the difference is computed pixel by pixel between center scale $c \in \{2, 3, 4\}$ and surround scales $s = c + \delta$ where $\delta \in \{3, 4\}$. Since the surround scale will by definition be smaller than the center scale, each surround scale is upscaled to the size of the center scale and then a pointwise difference operation is applied. This cross-scale differencing operation is denoted by $\Theta$.

The intensity channel CSD $I(c, s)$ is given by (1), where $I(c)$ is the intensity of the center scale $c$ and $I(s)$ is the intensity of the surround scale $s$

$$I(c, s) = |I(c) \Theta I(s)|.$$  

(1)

The color channel CSD $RG(c, s)$ is given by (2), where $RG(c)$ and $RG(s)$ are the red/green color opponency of the center and surround scales, respectively. Similarly, (3) describes the CSD for the blue/yellow BY color opponency channel

$$RG(c, s) = |RG(c) \Theta RG(s)|$$  

(2)

$$BY(c, s) = |BY(c) \Theta BY(s)|.$$  

(3)
The flicker channel CSD \( F(c, s) \) is given by (4), where \( F(c) \) and \( F(s) \) are the flicker of the center and surround scales, respectively
\[
F(c, s) = |F(c) \Theta F(s)|. \tag{4}
\]

The orientation channel CSD \( O(c, s, \theta) \) is given by (5), where \( O(c, \theta) \) and \( O(s, \theta) \) are the orientations of the center and surround scales, respectively, at the desired orientation \( \theta \)
\[
O(c, s, \theta) = |O(c, \theta) \Theta O(s, \theta)| \quad \text{s.t. } \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}. \tag{5}
\]

The motion channel CSD \( M(c, s, D) \) is given by (6), where \( M(c, D) \) and \( M(s, D) \) are the motions of the center and surround scales, respectively, at the desired direction \( D \)
\[
M(c, s, D) = |M(c, D) \Theta M(s, D)| \quad \text{s.t. } D \in \{\leftarrow, \rightarrow, \uparrow, \downarrow\}. \tag{6}
\]

\section*{C. Cross-Scale Addition}

Next, the feature pyramids are converted into conspicuity maps at channel granularity and they are listed in (7)–(11). Taking the color channel as an example, all its feature maps will be resized to the feature map size at Scale 4, the medium scale of Scale 0–8. Then the cross-scale addition (CSA) denoted as \( \oplus \), will be applied to the feature pyramids, with center \( c \in \{2, 3, 4\} \) and corresponding surround \( s = c + \delta \) where \( \delta \in \{3, 4\} \). A normalization step, denoted as \( \cdot() \) is applied to each input before addition. First, we find the global maximum \( M \) of the input and the average \( \mu \) of all other local maxima in that input. A pixel is considered a local maximum if its values exceeds all eight of its immediate neighbors. \( \mu \) is then the average of all such local maxima values. We then multiply the input by \( (M - \mu)^2 / M \) to normalize it
\[
\tilde{I} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{+4} \cdot(I(c, s)) \tag{7}
\]
\[
\tilde{C} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{+4} \cdot(\cdot(RG(c, s)) + \cdot(BY(c, s))) \tag{8}
\]
\[
\tilde{F} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{+4} \cdot(F(c, s)) \tag{9}
\]
\[
\tilde{O} = \sum_{\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}} \cdot(\oplus_{c=2}^{+4} \oplus_{s=c+3}^{+4} \cdot(O(c, s, \theta))) \tag{10}
\]
\[
\tilde{M} = \sum_{D \in \left\{\leftarrow, \rightarrow, \uparrow, \downarrow\right\}} \cdot(\oplus_{c=2}^{+4} \oplus_{s=c+3}^{+4} \cdot(M(c, s, D))). \tag{11}
\]

\section*{D. Saliency Map}

The last step involves computing the saliency map. In this step, all the five conspicuity maps from the color, intensity, orientation, flicker, and motion channels are normalized and summed to give a saliency map
\[
S = \cdot(\tilde{I}) + \cdot(\tilde{C}) + \cdot(\tilde{O}) + \cdot(\tilde{F}) + \cdot(\tilde{M}) \tag{12}
\]

\section*{IV. LED SYSTEM}

\section*{A. LED Backlight Types}

The LED backlighting technology adopted by industry can be further divided into two subcategories, as shown in Fig. 3.
of power can be saved by utilizing the proposed techniques. Power in a 2-D LED array and we show that a large portion can be further divided into zones for ease of control. The backlight panel has LED lights installed on the edges of the panel. With this installation, the edge-lit panel is cheap, slim, and has a low-power system overall. The other type is a 2-D array LED technology shown in Fig. 3(b). A backlight panel in this type will have a dense array of LEDs distributed across the whole panel. The LED array will be processed pairs and small buffers are used.

Fig. 3. LED panel types. (a) LED edge-lit panel. (b) LED array panel.

The first is an edge-lit technology shown in Fig. 3(a). As the name indicates, the backlight panel has LED lights installed on the edges of the panel. With this installation, the edge-lit panel is cheap, slim, and has a low-power system overall. The other type is a 2-D array LED technology shown in Fig. 3(b). A backlight panel in this type will have a dense array of LEDs distributed across the whole panel. The LED array will be further divided into zones for ease of control. The industry is moving from edge-lit panels to 2-D array panels for higher end displays due to superior image quality and better contrast. As a result, this paper focuses on how to save power in a 2-D LED array and we show that a large portion of power can be saved by utilizing the proposed techniques on zone dimming. Details of the 2-D array LED technology are shown in Fig. 4(a). An LED array panel with 576 LEDs is evenly distributed across the surface. In each zone, nine LEDs are embedded and chained together so that a single control unit can control all their luminance levels. In this paper, an industry standard HD LED backlight panel with 128 zones (8 × 16) is used, with each zone having nine LEDs [26]. All the experiments conducted in this paper are done based on this LED panel configuration.

B. LED Power Model

The power model from [27] is reused in this paper and the details are provided in

\[ P = 0.00374b^2 + 3.2194b - 10.576 \]  

where \( b \) is the average luminance in the zone and total power \( P \) is computed in milliwatt based on it. The model is calibrated carefully to meet the power consumption (100 W) of a typical 40-in LED TV working at full luminance level. The correlation between luminance level and power consumption is also evaluated and shown in Fig. 4(b). A noticeable power drop can be seen when the luminance level of backlight panel is reduced from 100% to 50%, which indicates that substantial power can be achieved via appropriate dimming.

V. SYSTEM ARCHITECTURE

There are three major components in our saliency-based adaptive LED panel system. The saliency core is responsible for finding the salient regions in the input image, and it has five channels—intensity, color, orientation, flicker, and motion—as indicated in Section III. We adopt a configurable generic channel architecture that can perform different tasks. The dimming core will dim the zone of the LED panel based on the saliency response passed from the saliency core. The compensation core takes care of color compensation of each pixel and sends it to the LCD panel. Two paths exist in our system as illustrated in Fig. 5. The first path, highlighted in red, involves the saliency core, which identifies the most salient regions and consequently the corresponding LED zones, and sets the luminance level of the zones that overlap with salient regions. The second path, highlighted in blue, performs pixel compensation based on the located salient regions.

A. Saliency Core

The saliency core takes in the input pixel stream and generates a saliency map by encoding biological responses to low-level pixel features. The saliency map is then projected on an 8 × 16 array based on the zone partition of the current LED backlight panel. The positions of salient zones will be passed to both, the dimming core and the compensation core.

In the core, the image preprocessing module serves as a first stage filter that decomposes the pixel stream into intensity \( I \) and color-opponency (RG/BY) streams for subsequent use. The second stage consists of the feature channels, each of which can be computed by the generic channel architecture. This pipeline is a configurable FPGA image processing macro containing six functional blocks as shown in Fig. 6 and is based on work by [21].

1) The first block is an image-differencing block. An image buffer of size 256 × 256 pixels is used to buffer the current frame. Simultaneously, a frame difference is computed with the corresponding pixels from the previous frame.

2) The second block is a pyramid generator, which produces nine scales (from 0 to 8) using the input image of size 256 × 256 as the base scale. Every subsequent scale is produced by low-pass filtering and downsampling the current scale. The downsampling factor is set to two. Thus the last scale is of size 1 × 1. In our design, the generator was wisely implemented through a 1-D downsampling block. When an image comes in as a stream of pixels, it will first pass the X-direction processing to downsample the input along the row dimension and then the Y-direction along the column dimension. The output stream is then buffered for later use. In total, eight \( X-Y \) processing pairs and small buffers are used. Between each pyramid level, a 5 × 5 Gaussian convolution filter is applied to avoid aliasing effects.
3) The third block is a Gabor filter block and it is used for generating edge features. In this saliency model, four orientations \{0°, 45°, 90°, 135°\} are needed; therefore, four dedicated Gabor filters are used.

4) A Motion filter block is the fourth block in the channel. It is a postprocessing step after the image pyramid is generated. In this step, a Reichardt motion detector is implemented to estimate the motion energy in four directions \{←, →, ↑, ↓\}.

5) The fifth block computes the CSD and normalization. In both, the center and the surround scales are ready from the pyramid generation block, the computation will kick in. Because the center scale is smaller than the surround scale, resizing it to the same size as the surround scale is required before computing. Instead of wasting addition storage and logic to do the resizing, the pixel position indices of the surround scale are used to locate the corresponding pixel in the center scale for differencing computing. The dimension ratio between the center and surrounding scale is fixed because it is known in advance which two scales are chosen for CSD computing. For example, when \(c = 2, s = 4\), the ratio is 1:4. The \(x\) and \(y\) positions of surround scale can be left shifted 2-bit to index the pixel of center scale. An example is shown in Fig. 7. Normalization is done in a streaming fashion. Global maximum and location maxima are computed via comparing logic with intermediate buffers. The final normalization coefficient \((M − \mu)^2 / M\) is available after the stream of differencing results between two scales is finished.

6) The sixth block is center-surround addition and normalization. The normalization processing performs the same task as described in the previous block. Because the CSA operation needs to merge all the center scales and surround scales to a medium scale with fixed size (Scale 4 with size 16 × 16), the resizing has to be done for both center and surround scales when they have different sizes. For the smaller size scale, the same method as shown in Fig. 7 is used for indexing. For the larger size scale, the \(X–Y\) downsampling pairs are used. After resizing, all the scales are added via an adder-tree to generate a conspicuity map.

7) The controller is responsible for the pipeline configuration. Before computing, a configuration package will be sent to the controller. Based on the information in the package, the controller will send the respective control signals to each compute module and set the dataflow of the pipeline channel. Two examples are shown in Fig. 8.
In the flicker channel [Fig. 8(a)], the frame-differencing block will be enabled and its output will be set as data input of pyramid generation block. Gabor and Reichardt filters are bypassed. In the motion channel [Fig. 8(b)], the frame-differencing block will be disabled and the Reichardt filter is used for motion detection.

The addition and averaging core will compute the final saliency maps from all the conspicuity maps of all enabled channels. The ranking core will find the $N$ most salient regions from the map and send it to the corresponding LED zone on the backlight panel.

**B. Dimming Core and Dimming Strategy**

The dimming core is used to control the backlight panel. The saliency map, when ready from saliency core, will be used to set the luminance value of the corresponding LED zone. First, top $N$ salient regions in the map will be chosen based on their saliency value. Then, the saliency map will be projected to the backlight panel to locate the corresponding $N$ LED zones that have the largest overlapping portion. Matched LED zones, also called salient zones, will be set to a new luminance value $L_{\text{initial}}$, which is determined by scaling the original luminance by a predetermined fraction. A fading range $d_{\text{fading}}$ is also defined to help determine the luminance value of other LED zones. If an LED zone is located within the fading range of its closest salient zone, its luminance value will be scaled down linearly as a function of its distance from that salient zone. The luminance of a zone, $L_z$, is calculated using (14) where a scaling factor $\gamma$ is used for linear scaling.

$$L_z = L_{\text{initial}} - \gamma \times d_{\text{ToCenter}} \quad \forall d_{\text{ToCenter}} \in [0, d_{\text{fading}}].$$

Fig. 9(a) and (b) shows the ideal luminance setting when a single (two) salient LED region(s) is (are) located at grid point $[5, 5]$ (grid points $[4, 9]$ and $[5, 5]$). Fig. 9(c) shows the luminance distribution of (b) after diffusing. The parameters used here are: 1) $L_{\text{initial}} = 80\%$; 2) $\gamma = 10\%$; and 3) $d_{\text{fading}} = 3$. LED zones on the backlight panel are represented as small boxes. The shades are used to indicate the zones’ luminance value. The brighter the box, the higher the luminance value of the box and vice versa.
The influence of these three parameters, $L_{\text{initial}}$, $\gamma$, and $d_{\text{fading}}$, is studied via two experiments. We first varied $L_{\text{initial}}$ and $\gamma$ while keeping the fading distance constant at a value of 3. Using (13) and (14), we calculate the power saved. As shown in Fig. 10(a), a high $L_{\text{initial}}$ value or many salient regions can result in a loss of power saving. For a small number of salient regions, a larger scaling factor $\gamma$ can provide more potential in power saving since the nonsalient LED zones get more dimming. This effectiveness diminishes when the number of allowed salient regions increases. In Fig. 10(b), the scaling factor $\gamma$ and initial luminance $L_{\text{initial}}$ are fixed to 10% and 80%, respectively, while the fading range $d_{\text{fading}}$ is changed. A similar diminishing behavior is observed in this case too. Based on these discoveries, the values of $L_{\text{initial}}$, $\gamma$, and $d_{\text{fading}}$ are set to 80%, 10%, and 3, respectively, for all the image experiments in this paper. Also two salient regions are allowed in one frame for our evaluation. The settings of the system for video test cases use the same concept presented here but with different values and they are explained in Section VI-C.

### C. Compensation Core

There are three major factors to determine how the luminance of a picture on an LCD display is perceived by the human eye: 1) emitted luminance of the backlight; 2) transmittance of the liquid crystal in the LCD panel; and 3) image intensity $Y$. The final perceived luminance is the product of multiplying them together as shown in (15) and is conceptually shown in Fig. 11. It should be noted that we use the luminance output of the preprocessing step of saliency as the image intensity here

$$L_{\text{perceived}} = L_{\text{emitted}} \times \rho \times Y. \tag{15}$$

As indicated in the equation, when the backlight is dimmed, the other two factors, transmittance and image intensity, can be modified accordingly to keep the perceived luminance unchanged. Because the salient zones of our proposed system are dimmed to 80% of the original luminance, the image compensation coefficient is set to 1.2 for all the pixels within the salient zone. For the nonsalient LED zones, the compensation coefficient is set based on their distance to the closest salient zone. It is similar to the method used in the luminance dimming strategy but in a reversed way. As a result, the image compensation coefficients are set to 1.2, 1.3, 1.4, and 1.5 from the center to boundary. And the new perceived luminance is 96%, 91%, 90%, and 85% for the case when the fading range is equal to 3.

Fig. 12(a) illustrates the compensation coefficients of the neighboring LED zones around the salient zone with distances of 1, 2, and 3 region(s) away. In Fig. 12(a), many sharp changes can be noticed at the edges between two salient zones because the perceived luminance also changes dramatically at the edges. To smooth the edge artifact, an inverse-Gaussian
TABLE I
RESOURCE UTILIZATION ON VIRTEX 6 SX475T

<table>
<thead>
<tr>
<th>Slice Regs</th>
<th>LUTs</th>
<th>BRAM36</th>
<th>BRAM18</th>
<th>DSPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>167258 (28%)</td>
<td>184734 (62%)</td>
<td>589 (55%)</td>
<td>399 (18%)</td>
<td>807 (40%)</td>
</tr>
</tbody>
</table>

Table showing resource utilization on a Virtex 6 SX475T FPGA.

Fig. 13. Saliency core running on a Virtex6 FPGA. Image is captured via a camera connected to the host CPU and transferred to the core via PCIe.

Function is used for a pixel level compensation. Instead of having all the pixels in a zone set to the same coefficient, each pixel will get a compensation coefficient. The center pixel of the salient zone will get the lowest compensation value 1.2, while the pixels beyond the fading range will be compensated with largest value at 1.5, because they are dimmed the most. The finer granularity compensation results are shown in Fig. 12(b) for comparison. A much smoother transition can be seen at the edge of salient zones. It should be noted that the coefficients are symmetrical in a zone and hence only a quarter of them need to be saved in memory.

VI. EXPERIMENTS AND RESULTS

A. System Constraints

For prototyping purposes, our saliency core is implemented with one generic channel on a Virtex 6 FPGA. Saliency is computed iteratively for each of the five channels. Configuration packages are sent into the core before a particular channel computation starts. The overall throughput of this core is 30 FPS for a 512 × 512 image frame. Since our on-chip image buffer can hold up to 256 × 256 pixels, we divide the incoming frames into chips of size 256 × 256 and process these chips in a streaming fashion. The resource utilization for this design is shown in Table I.

To estimate the power overhead of the saliency-based system, we first estimated the power consumption using Xilinx Xpower, which was around 12 W. Next, we verified this number by measuring power on-board using a power meter. Fig. 13 shows the saliency core running at 100 MHz on a single Virtex 6 FPGA of an FPGA cluster. When we ran the saliency core, we obtained a delta power (running power—idle power) of 14 W, thus validating our analysis. Though this power may be prohibitive for deployment in many displays, an ASIC implementation is likely to fit within the display’s power budget. Given that the dynamic power ratio between FPGA and ASIC is approximately 14 times [28], the additional power budget needed to accommodate an ASIC would be ~1 W. Therefore, the system built in ASIC only occupies 1%–2% of the total power that is consumed by TV-scale (30+ in) LCD display systems.

As we are targeting HD video-based applications, we project our experimental results to a saliency core with all five channels running in parallel. The image preprocessing module remains common to all channels. Each generic pipeline will be configured according to the assigned function. In the orientation channel, the Gabor filtering block will be enabled. Similarly, the Reichardt filtering block is required only for the motion channel. In the flicker channel, the frame-differencing block is turned on for computing the pixel differences between two continuous frames. For the color and intensity channels, the differencing, Gabor, and Reichardt blocks are all off because they are not needed here. The resource utilization of such a saliency system is estimated based on a rough linear projection and it can be easily mapped to a DINI-Group DNV7F2B board [29] with a 5× performance improvement and 5× power consumption. While a 5× power consumption may seem too conservative given a technology node change (40–28nm), we compensate for the fact that in order to achieve real-time performance for HD video we will have to run the core at a slightly higher frequency than 100 MHz to achieve the same throughput as the one running on-board. The main component in the compensation core is a Gaussian core which stores all the coefficients used for color compensation. It can be easily mapped to the block rams of a Virtex 7 FPGA.

B. Image Distortion Verification

The potential change of salient regions between the original images and the compensated image is evaluated in this section. For quantitative metrics, the Euclidean distance is computed between the most salient regions of original images and compensated images based on the zone partition of the backlight panel (8 × 16 in our case). For example: if the most salient region is [4, 5] and [4, 6] for the original and compensated image, respectively, then the distance is one. Further, changes of salient regions are categorized into three classes: 1) Same, if the distance is 0; 2) Shift, if the distance is 1 or 2; and 3) New, if the distance is above 2. The result shows the most salient regions remain at the same position for 95% of all the test images, with 4% causing small translational shift without noticeable distortion. For only around 1% of the cases, the compensated image introduces the salient region to a new location. For such outlier images, a user experience test was conducted. 15 candidates were randomly picked and shown 20 image pairs (original and compensated). They were asked to give their perception of the image pairs from three choices: different, dimmed, or same.

The test results are shown in Table II. As indicated, visual perception of compensated images was not significantly distorted. Most of the image pairs look the same or dimmed. Several images are randomly picked and shown...
Fig. 14. Image examples after compensation. (a) Beach picture with LED zone [6, 3], [2, 15]. (b) Mountain picture with LED zone [8, 2], [8, 3]. (c) Mountain picture with LED zone [3, 3], [7, 15]. (d) Moon picture with LED zone [4, 2], [4, 12].

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>USER EXPERIENCE TEST ON IMAGE DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio</td>
<td>Different</td>
</tr>
<tr>
<td></td>
<td>4.33%</td>
</tr>
</tbody>
</table>

TABLE III | IMAGE QUALITY METRICS |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min.</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.84</td>
</tr>
<tr>
<td>MSE</td>
<td>0</td>
</tr>
</tbody>
</table>

in Fig. 14(a)–(d) for a qualitative assessment of image distortion. The sub-pictures (i), (ii), (iii), and (iv) represent the original image, compensated image, dimmed image and salient map, respectively. With the help of color compensation, no significant difference can be noticed between (i) and (ii).

To further quantify the similarity between the original image and the compensated image, we used a popular benchmark used to evaluate saliency models [30]. We ran our saliency-driven compensation model on this data set which consists of 1003 images. We then evaluated the mean square error (MSE) and the measure of structural similarity (SSIM) between the original image [Fig. 15(a)] and the compensated image [Fig. 15(b)]. The MSE is a common metric used in signal processing to compute the error between the reference and the test signal. An MSE score of 0 indicates that there is no error between the two. In terms of human perception, the SSIM is a popular metric proposed in [31] that combines luminance, contrast and structure for evaluating image quality based on the human visual system. The SSIM index varies between 0 and 1, where 1 indicates that the quality of the test image is the same as that of the reference image. We quantify the minimum, maximum, average, and standard deviation for both these metrics when evaluated on the entire data set of 1003 images and tabulate the results in Table III. The average MSE score is close to 0, while the average SSIM score is close to 1, indicating high image quality after compensation. We also show the SSIM map obtained from MATLAB in Fig. 15(c) as an example, where areas of high intensity indicate high similarity.

C. Video Distortion Verification

When buying a TV, people consider two kinds of image quality: one is the single-static-image quality and the other is the image quality in a video. Our saliency power saving system is also verified to see whether it can maintain the perceived visual quality when it is applied to video clips.

All the system settings remain unchanged from the image test. While evaluating our model on videos, a shimmer effect was observed when a salient object was in motion. This is because, for a similar salient object, if its Euclidean distance in consecutive frames is greater than 2, which also means a
large distance jump of the salient zone in the backlight panel, people begin to notice the luminance zone changes in the compensated video. Due to this reason, salient zones in the previous frame are only allowed to shift toward their neighbors vertically, horizontally or diagonally in the current frame. With this constraint, smooth salient region shifts for slow moving objects can be achieved. No particular optimizations are done for fast moving objects because they will be out of the camera view within several frames.

However, this method does not eliminate all shimmers that can occur in the video. Let us consider a very simple scenario: a yellow bottle is kept on a gray-color table with another red bottle on its right side with some distance (out of the camera view when the camera is focused on the yellow bottle). The recording is started with the camera facing the yellow bottle. When the camera turns its head slowly toward the red bottle, a new salient region will be introduced because low-level features like color and intensity are activated as long as the red bottle is in the camera’s sight. Since the new salient region in the current frame is too far away from the salient region (yellow bottle) in the previous frame, the new salient region will move toward the red bottle within several frames because of the movement constraints introduced. As a result, in the final compensated video, people will notice a luminance trace from the yellow bottle to red bottle. To solve this problem, the scaling factor $\gamma$ is set to 5% rather than 10% to reduce the luminance difference between the salient zone and its surrounding zones. Correspondingly the compensation coefficients are adjusted to 1.2, 1.25, 1.3, and 1.35.

As a user perception test, a pair of videos, the original video and the compensated video, were randomly picked from a set of 29 videos and shown to ten candidates. No one noticed the shimmer in the compensated video. It should be emphasized that removal of the shimmer does not introduce any other new artifacts. Fig. 16 shows a qualitative example from our video compensation. Here, Fig. 16(a)–(d) represents the original frame, compensated frame, dimmed frame, and saliency map, respectively.

D. Power Comparison

LCD power consumption of our system is evaluated on both image and video displaying cases. In addition, the power consumption is also evaluated via methods presented in [15] and [16] for comparison. The same 3000 HD images and 29 videos are tested with each method and the result is depicted in Fig. 17. In the static image testing, the highest saving ratio is achieved by our system. But in the video test, our method loses to the method in [15] due to the fact that a smaller scaling factor for dimming is chosen to preserve the video quality.

It should be noted that the $L_{\text{initial}}$, described in Section V-B, is the luminance value that is assigned to the salient zones. It can be set to a lower value of 70% of the full luminance for more power saving. Therefore, our system can at least reach the same power saving as [15] does. With the help of image compensation, not implemented in [15], more power saving can be achieved in our system with very little sacrifice of the image quality. Last but not the least, the extended saliency system gives the lowest standard deviation of power saving under both image and video tests as shown in Fig. 18. As discussed earlier in Section II, active backlight dimming methods always have a wide range of power saving

![Image examples from [30] after compensation. (a) Original image. (b) Compensated image. (c) SSIM map.](image_url)
from 20% to 80%. This is because the active dimming methods are sensitive to the image contents like the average pixel intensity value. As a result, they fail to achieve the desired power saving when images/videos have bright background. On the contrary, the saliency model adopted in our system uses low-level features from images and videos to locate the potential attention region. With appropriate number of salient zones allowed in the backlight panel, this algorithm insures our system can always provide concrete power savings regardless of display content.

VII. CONCLUSION

This paper adopts and modifies an FPGA accelerator macro composed of a set of basic image processing primers and aims at providing a flexible design space for developing custom vision accelerators. Based on this generic channel architecture, we implement a biologically-inspired pixel-driven priority scheme for lighting intensity to reduce the power consumption of LCD display systems. Our methodology dims the LED backlight zones at real-time speed based on the salient regions found via the saliency accelerator.
We ran a user perception test and also quantify the distortion using two popular metrics: SSIM and MSE. Our scheme is data-driven, and hence the distortion error will vary from frame to frame. However, we quantify the variation on a public data set consisting of 1003 images. [16] uses a fixed MSE threshold of 0.4, but they also highlight the importance of having a variable threshold based on image size. Also, their method involves an iterative process to achieve the predefined MSE threshold while our method is a feed-forward scheme requiring no tuning. [15] does not provide any distortion detail to compare against. In the image test with 3000 random HD pictures, the proposed systems can achieve an average power saving of 66%, which is the best among all the active dimming systems. Although no obvious artifacts can be found on the compensated image with the original parameters, a shimer effect is found on the compensated video. To solve this problem, new parameters and movement constraints of salient zones are proposed. With these new settings, our system can achieve a concrete power saving of 52% with minimal standard deviation. A video quality validation using 29 videos is also conducted with ten different test subjects. Based on these test results, it can be concluded that no difference is observed between the compensated video and the original video. It should be noted that more power saving can be reached with very little video quality degradation if more dimming is applied on the salient zones in quality-of-service-based applications.

REFERENCES


Yang Xiao received the B.E. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2005, the M.S. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 2008, and the Ph.D. degree in computer science and engineering from Pennsylvania State University, State College, PA, USA, in 2014. He joined Synopsys, Mountain View, CA, USA, in 2014, where he is currently a Senior Research and Development Engineer. He is involved in physical optimization related research.

Siddharth Advani received the B.S. degree in electronics engineering from Pune University, Pune, India, in 2005, and the M.S. degree in electrical engineering from Pennsylvania State University, State College, PA, USA, in 2009, where he is currently pursuing the Ph.D. degree in computer science and engineering. His current research interests include embedded vision design for smart wearables and reconfigurable computing for domain-specific applications.
Donghwa Shin received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in computer science and electrical engineering from Seoul National University, Seoul, Korea, in 2005, 2007, and 2012, respectively.

He is currently an Assistant Professor with the Department of Computer Engineering, Yeungnam University, Gyeongsan, Korea. His current research interests include system-level low-power techniques for embedded system and hybrid power system design.

Naehyuck Chang (F’12) received the B.S., M.S., and Ph.D. degrees from the Department of Control and Instrumentation, Seoul National University, Seoul, Korea, in 1989, 1992, and 1996, respectively. He was with the Department of Computer Science and Engineering, Seoul National University, from 1997 to 2014. He was an LG Yonam Foundation Research Professor in 2005. He served as a Vice Dean of the College of Engineering with Seoul National University from 2011 to 2013. He has been a Full Professor with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea, since 2014. His current research interests include low-power embedded systems and design automation of things, such as systematic design and optimization of energy storage systems and electric vehicles.

Dr. Chang is a fellow of the Association for Computing Machinery (ACM). He was a recipient of several International Symposium on Low Power Electronics and Design (ISLPED) low-power design contest awards, the 2012 SAE Vincent Bendix Automotive Electronics Engineering Award, the 2014 ISLPED Best Paper Award, and the 2009 IEEE SSCS Seoul Chapter Award. He was the Chair of the ACM Special Interest Group on Design Automation. He is the Editor-in-Chief of the ACM Transactions on Design Automation of Electronics Systems, and an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. He also served as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE EMBEDDED SYSTEMS LETTERS, and ACM Transactions on Embedded Computing Systems. He was the General Co-Chair of VLSCI-SoC, the International Conference on Computer Design (ICCD) in 2014 and 2015, and ISLPED in 2011. He is the Technical Program Co-Chair of the Design Automation Conference, and was the Technical Program Co-Chair of the Asia and South Pacific Design Automation Conference in 2015, ICCD in 2014, CODES + ISSS in 2012, and ISLPED in 2009.

John (Jack) Sampson received the B.S. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2002, and the Ph.D. degree in computer engineering from the University of California at San Diego, San Diego, CA, USA, in 2010.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Pennsylvania State University, State College, PA, USA. His current research interests include energy-efficient computing, architectural adaptations to exploit emerging technologies, and mitigating the impact of dark silicon.

Vijaykrishnan Narayanan (F’11) received the B.S. degree in computer science and engineering from the University of Madras, Chennai, India, in 1993, and the Ph.D. degree in computer science and engineering from the University of South Florida, Tampa, FL, USA, in 1998.

He is currently a Professor of Computer Science and Engineering and Electrical Engineering with Pennsylvania State University, State College, PA, USA. His current research interests include power-aware and reliable systems, embedded systems, nanoscale devices and interactions with system architectures, reconfigurable systems, computer architectures, network-on-chips, and domain-specific computing.

Dr. Narayanan is a fellow of the Association for Computing Machinery (ACM). He has received several awards, including the Penn State Engineering Society Outstanding Research Award in 2006, the IEEE CAS VLSI TRANSACTIONS Best Paper Award in 2002, the Penn State CSE Faculty Teaching Award in 2002, the ACM Special Interest Group on Design Automation Outstanding New Faculty Award in 2000, the Upsilon Pi Epsilon Award for academic excellence in 1997, the IEEE Computer Society Richard E. Merwin Award in 1996, and the University of Madras First Rank in Computer Science and Engineering in 1993. He has received several certificates of appreciation for outstanding service from ACM and the IEEE Computer Society. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.