Low-Overhead Adaptive Brightness Scaling for Energy Reduction in OLED Displays

Low-Overhead Adaptive Brightness Scaling for Energy Reduction in OLED Displays

Daniele Jahier Pagliari, *Member, IEEE,* Santa Di Cataldo *Member, IEEE,* Edoardo Patti, *Member, IEEE,* Alberto Macii, *Senior Member, IEEE,* Enrico Macii, *Fellow, IEEE,* and Massimo Poncino, *Fellow, IEEE*

Abstract—Organic Light Emitting Diode (OLED) is rapidly emerging as the mainstream mobile display technology. This is posing new challenges on the design of energy-saving solutions for OLED displays, specifically intended for interactive devices such as smartphones, smartwatches and tablets. To this date, the standard solution is brightness scaling. However, the amount of the scaling is typically set statically (either by the user, through a setting knob, or by the system in response to predefined events such as low-battery status) and independently of the displayed image.

In this work we describe a smart computing technique called Low-Overhead Adaptive Brightness Scaling (LABS), that overcomes these limitations. In LABS, the optimal content-dependent brightness scaling factor is determined automatically for each displayed image, on a frame-by-frame basis, with a low computational cost that allows real-time usage.

The basic form of LABS achieves more than 35% power reduction on average, when applied to different image datasets, while maintaining the Mean Structural Similarity Index (MSSIM) between the original and transformed images above 97%.

Index Terms—Displays, OLED, Energy Efficiency, Smart Computing, Approximate Computing

1 Introduction

Interactive devices such as smartphones, smartwatches and tablets with increasingly larger and higher resolution displays are an emerging trend in the consumer electronic market; since displays are known to be among the most power-hungry components, this trend has an important impact on the power consumption of these devices.

One solution to tackle this issue has been technological: Organic Light Emitting Diode (OLED) panels are rapidly emerging as the mainstream mobile display technology and progressively replacing classic Thin Film Transistor (TFT) LCDs [1]. Besides higher power efficiency, they also provide higher brightness, better viewing angles, and the possibility of building thinner and flexible panels [2].

However, the *emissive* nature of OLEDs (i.e., the fact that their power consumption strongly depends on pixel values) does not necessarily translates into a power benefit. While generally more efficient than LCDs, OLEDs consume significantly more power for bright images [1]. Therefore, just changing the display technology only partially addresses the power consumption issue, and proper OLED *power management strategies* have become a critical design requirement. In response to this need, a wide body of methods for reducing power in these displays have been proposed, most of which exploit the aforementioned image-dependence feature.

The most popular solution is based on applying a *transformation* to the image shown on the panel, thus trading off the achievable saving and the "fidelity" of the content [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Most transformations consist of *mappings of pixel intensities*, i.e tabular or analytical functions linking pixel intensity x in the input image to a corresponding value $T(x)$ in the output image. The various proposed methods differ in the shape of the mapping function $T()$ and in the possibility of $T()$ changing *dynamically* at runtime depending on the input image.

Among these virtually infinite mappings, the simplest one to achieve power saving is $T(x) = kx$, i.e. a scaling of each pixel intensity by a factor $k < 1$. This is the typical implementation of the "brightness" knob in OLED devices, and also the action taken by most systems in response to an event requiring a reduction in the power consumption (e.g., low battery).

Although simple and effective in terms of power reduction, this standard *brightness scaling* solution has the major drawback of being *image-agnostic*. In order to reflect the sensitivity of OLED power to pixel intensities, a more effective solution should adapt the brightness scaling factor to the power consumption of the current image, i.e., brighter images should be scaled more with respect to darker ones (for which the resulting power reduction would be negligible).

In this paper, which extends our previous work of [16], we present a smart brightness scaling approach that addresses these issues. Our technique, called *Low-overhead Adaptive Brightness Scaling* (LABS) optimizes the tradeoff between power reduction and alteration for each displayed image, using a combined power-similarity metric. The optimal scaling factor for the target image can be derived from this metric either analytically or through a regression-based approach. While the first solution is exact the second is approximate. However, it allows to estimate the optimal fac-

D. Jahier Pagliari, S. Di Cataldo, E. Patti, A. Macii and M. Poncino are with the Department of Control and Computer Engineering, Politecnico di Torino, Turin, IT. E. Macii is with the Interuniversity Dept. of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, IT.

E-mail: {name.surname}@polito.it

Manuscript received January XX, XXXX; revised January XX, XXXX.

tor in real time, with small software/hardware overheads, despite yielding a very small error compared to the exact method.

With respect to [16], this work presents several new insights and extensions of the LABS technique, including:

- A new formulation of the optimization metric used to extract the optimal scaling factor for an image, which allows to set a user-determined "power reduction effort", hence allowing runtime configuration of the power versus image similarity tradeoff.
- A detailed analytical formulation of the optimal scaling factor computation problem and of its exact solution.
- A detailed analysis of the software and hardware complexity of LABS.
- Completely new experimental results, now including the comparison among different regression models, between the exact and approximate scaling factors derivations, etc.

In its basic form, LABS obtains an average power saving greater than 35% for different reference image datasets, while keeping a very high similarity between input and output images (97% on average, using the MSSIM metric [17]).

The rest of the paper is organized as follows. Section 2 introduces the required background on OLEDs and presents related work. Section 3 describes the theoretical foundation of our proposed method, while Section 4 presents the details of its implementation. Section 5 presents some experimental results and Section 6 concludes the paper.

2 Background and Related Work

Unline traditional LCDs, which rely on a backlight source, OLEDs are *emissive*, i.e. light is directly produced by panel pixels. The latter are formed by a combination of three devices, emitting light for the three RGB channels [3]. Each color intensity depends nonlinearly on the current flowing through the corresponding device; therefore, the total display power is strongly affected by pixel intensities. Measurements on real OLED panels allowed to build an empirical power consumption model [3] :

$$P_{tot} = \sum_{i=0}^{W}\sum_{j=0}^{H}(w_0 + w_r \cdot R_{i,j}^{\gamma} + w_g \cdot G_{i,j}^{\gamma} + w_b \cdot B_{i,j}^{\gamma}) \quad (1)$$

where W and H are the width and height of the panel, $(R_{i,j}, G_{i,j}, B_{i,j})$ are the sRGB components of the pixel at position (i, j), and w_x and γ are panel-dependent coefficients, obtained via characterization [3]. Typically, $2 \leq \gamma \leq 3$.

Power reduction techniques for OLEDs are tailored to this image-dependent power model. Its most important consequence is that power consumption can be altered simply by *transforming pixel intensities in the displayed image*. These transformations do not require any type of hardware control, and can be implemented fully in software. In general, literature approaches can be split in two groups: those targeting *Graphical User Interfaces* (GUIs), and those for general images.

Techniques for GUIs are based on the observation that, for an interface, *usability* is more important than visual *fidelity*: ensuring that the user is able to distinguish text or icons from the background is more important than preserving the exact colors of the GUI. Following this principle, the authors of [4] propose a method that dramatically alters the color palette of a GUI, exploiting the fact that the three emissive devices in OLEDs have different power consumptions. This idea is extended in [3], where a web browser able to optimize OLED power consumption is proposed. A different GUI solution is proposed in [5], based on selectively applying brightness scaling to the pixels outside of the area of "user interest" (e.g. the foreground window, the selected list item, etc.). A more advanced implementation of same idea is found in [6], where the authors automatically assign a "degree of user attention" to each display region using *saliency maps*. Pixels are then dimmed depending on this measure, with a progressive scaling factor that avoids the creation of artifacts. Finally, a more recent approach targeting an energy-friendly Android app GUIs is described in [7], where a multi-objective optimization approach is proposed, balancing energy reduction with image contrast. Power reduction techniques for GUIs are not suitable for general images or videos, for which visual fidelity is a fundamental aspect of perceived quality and drastic modifications of color or brightness would be unacceptable.

An interesting solution for general images, proposed in [8], consists in applying *Dynamic Voltage Scaling* (DVS) to the OLED panel, thus reducing the maximum current that can flow through pixel devices and consequently the total display consumption. However, DVS also reduces the brightness of the displayed image. Therefore, similarly to backlight-scaling techniques for LCDs, brightness reduction is then compensated in software by an image transformation. An extension of this idea to allow multiple and independently voltage-scaled regions of a panel is described in [9].

DVS-based methods require custom display driver circuitry and cannot be applied to off-the-shelf OLEDs. Therefore, many researchers have proposed techniques that rely *solely on pixel transformations*. Starting from the observation that chrominance variations in a photo or video strongly alter the perceived quality, they typically act only on the luminance of pixels. Some approaches try to concurrently reduce power consumption and *enhance* the target image through non-linear transformations [10], [11], [12], [13], [22]. A fundamental technique in this category is *Power-Constrained Contrast Enhancement* (PCCE), first described in [10]. PCCE computes a nonlinear luminance mapping for each image, using an objective function that concurrently favors the reduction of power and the enhancement of contrast in the output. Many variants of PCCE have been proposed in literature; the solution in [11] uses multi-scale retinex to improve output quality, while [12] devises a similar transformation that does not require an iterative optimization procedure for extracting the luminance mapping. Transformations with similar objectives to PCCE, but adopting different algorithms are described in [13] and [22].

In all these solutions, identifying the parameters of the transformation and then applying it requires complex computations, that cannot be realistically implemented in real time for each new displayed frame. In fact, the refresh rate of displays requires this transformation to be repeated every

$\approx 15-20ms$ (50–60 Hz rate). Complex transformations such as those of [10], [11], [12], [13], [22] can only be implemented in such a short time by a powerful processors or a GPU, an option that implies very large energy overheads, that can offset the savings obtained on the display.

In response to this observation, some authors have devised low-overhead image transformation methods for OLEDs power reduction [14], [15], [23]. Specifically, in [14] and [15] overheads are reduced by moving most of the computational effort to the *image acquisition phase*, using a custom camera application for mobile devices. These solutions exploit the fact that the acquisition phase does not have the same real-time constraints of the display; however, they are applicable only to images are acquired through the provided camera application, and cannot be applied to downloaded or synthetic images. Another low-overhead solution is presented in [23], where a non-linear transformation that yields output quality comparable to PCCE is built through an offline training process, in which image features are put in relation to transformation parameters.

The most suitable image transformation for real-time usage is brightness scaling, which is therefore the common solution adopted in consumer devices nowadays [16], [24], [25]. For grayscale images, brightness scaling transforms every pixel with the function:

$$z = k \cdot x \tag{2}$$

where x is the pixel value and k is called *scaling factor*. For color images, the same effect is obtained scaling luminance components. To do so, pixels are first converted to an appropriate color space, such as YCbCr, where the Y component represents luminance. The conversion between RGB and YCbCr spaces is obtained through linear transformations:

$$[Y_{i,j}, Cb_{i,j}, Cr_{i,j}]^T = \mathbf{C} \cdot [R_{i,j}, G_{i,j}, B_{i,j}]^T + [0, 128, 128]^T \tag{3a}$$

$$[R_{i,j}, G_{i,j}, B_{i,j}]^T = \mathbf{D} \cdot [Y_{i,j}, Cb_{i,j}, Cr_{i,j}]^T - [0, 128, 128]^T \tag{3b}$$

where $\mathbf{C}, \mathbf{D} \in \mathbb{R}^{3x3}$ are matrices of constant coefficients. Using the JPEG standard for YCbCr conversion [26], both RGB and YCbCr pixels are represented on 24-bit, with each component spanning the range $[0, 255]$.

The effectiveness of brightness scaling in reducing OLED power consumption is given by the fact that $P_{tot} \propto Y^{\gamma}$. However, as mentioned in Section 1, brightness scaling is generally *image-agnostic*, and uses the same scaling factor for all images. The authors of [25] and [24] first investigated *adaptive* brightness scaling approaches. In both works, the proposed transformations minimize power consumption under a minimum quality constraint, using a *different scaling factor* for each image. Specifically, in [25] only brightness values higher than a minimum knee-point are scaled, and the acceptable quality reduction is quantified in terms of maximum Mean Squared Error (MSE) between input and output images. The work in [24], conversely, uses a uniform scaling for all brightness values, and its quality constraint is specified as a minimum threshold on the Mean Structural Similarity Index (MSSIM) [17] between input and output, which correlates better with the subjective perception of image alteration compared to the MSE. Although effective, these methods are still not suitable for an implementation in real time. Indeed, the whole transformation process takes several seconds for a 1080p image [24], [25].

While our work is inspired by the work of [24], LABS differs from [24] in the objective function used for calculating the parameters of the transformation and, most importantly, it simplifies the search for these parameters greatly, thus allowing runtime implementation.

3 Theoretical Foundation

3.1 Objective

Low-Overhead Adaptive Brightness Scaling (LABS) is, to the best of our knowledge, the first method to allow real-time computation of an image-dependent brightness scaling factor with low overheads. The final objective of our method is to replace the traditional brightness control knob present in most devices with a new, “power reduction effort” knob through which the user (or the operating system) can set the desired power reduction, at the cost of worsening the visual quality of the displayed image. Depending on this parameter, LABS automatically computes the scaling factor for each image.

When activated manually, this smart knob will allow a user not just to set a brightness level based on visual quality, but it will also provide clear (yet possibly qualitative) information on the expected power reduction (e.g. standard, half or double).

In order to achieve this objective, scaling factors in LABS are selected so to maximize an objective function which is the combination of two contrasting metrics, i.e. display power reduction and image similarity. This is different from previous solutions, in which one of the two metrics (typically power) was used as objective, while the other was used as constraint [24], [25]. Using a combined objective function allows to determine in a fully automatic way the scaling factor that yields the best tradeoff among similarity and power, given the desired power reduction effort. We construct such function starting from an observation on the effects of brightness scaling on image similarity, detailed in the next section.

3.2 MSSIM and Brightness Scaling

The Structural SIMilarity (SSIM) index [17] is a popular image similarity metric, that has been shown to correlate well with human perception. Its expression is as follows:

$$SSIM(x, z) = \frac{(2\mu_x\mu_z + c_1)(2\sigma_{xz} + c_2)}{(\mu_x^2 + \mu_z^2 + c_1)(\sigma_x^2 + \sigma_z^2 + c_2)} \tag{4}$$

In (4), x and z are subsets of the image pixels, obtained via a sliding window; μ_x, μ_z and σ_x^2, σ_z^2 are the mean and variance of pixels gray levels in x and z, and σ_{xz} is their covariance. The coefficients c_1 and c_2 are constant, and are added for stabilization purposes, i.e. to avoid $0/0$ divisions (e.g. when both images are totally black, $\mu_x = \mu_z = 0$).

In the case of color images, the $SSIM$ is computed replacing gray levels with the luminance component of each pixel, which can be extracted from YCbCr space using (3a).

$SSIM$ values for all positions of the sliding window are then averaged to compute the global *Mean SSIM* ($MSSIM$).

The result is a single real number in $[0 : 1]$, where $MSSIM = 1$ corresponds to identical images.

The authors of [24] have noted that (4) can be simplified when the pixels in z are brightness scaled versions of those in x, obtained with (2). Specifically, the following relations hold:

$$\mu_z = k\mu_x,\ \sigma_z = k\sigma_x,\ \sigma_{xz} = k\sigma_x^2 \tag{5}$$

Consequently (4) can be rewritten as:

$$SSIM(x, kx) = \frac{(2k\mu_x^2 + c_1)(2k\sigma_x^2 + c_2)}{[(1+k^2)\mu_x^2 + c_1][(1+k^2)\sigma_x^2 + c_2]} \tag{6}$$

What the authors of [24] did not consider, however, is that this equation can be further simplified by introducing a small approximation. Indeed, since c_1 and c_2 are only present in the equation for stabilization purposes, their value is chosen to be *very small* compared to the other terms. Standard choices are: $c_1 = (0.01L)^2$ and $c_2 = (0.03L)^2$, where L is the maximum pixel value, e.g. $L = 255$ in JPEG YCbCr [26]. It is easy to see that, with the pixel statistics of typical images, the error introduced by removing these two constants from (6) is very small. Doing so, the $MSSIM$ expression reduces to:

$$SSIM(x, kx) \approx \frac{(2k\mu_x^2)(2k\sigma_x^2)}{[(1+k^2)\mu_x^2][(1+k^2)\sigma_x^2]} = \frac{4k^2}{(1+k^2)^2} \tag{7}$$

The fact that (7) only depends on k, and not on the pixels in x means that the SSIM (and consequently also the MSSIM) obtained after a brightness scaling transformation is *approximately independent on the image*. This is empirically confirmed by the box plots in Figure 1, which show the distribution of MSSIM values between original and brightness-scaled images with $k = 0.8$, computed over three different image datasets [27], [28], [29]. As shown, most of the images have MSSIM values differing by less than 0.5%.

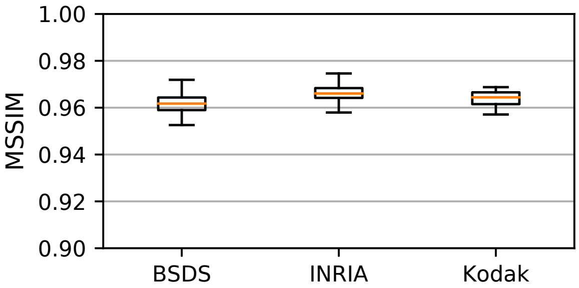


Fig. 1. MSSIM distribution for three different image datasets after brightness scaling with $k = 0.8$.

This apparently counter-intuitive result makes sense when considering that the SSIM/MSSIM is a metric of image similarity, and *not* of absolute image quality. Brightness scaling with constant k does not introduce any content-dependent distortion, so it is reasonable that the resulting similarity with respect to the original image is the same regardless of the content. From a practical point of view, this finding translates into the possibility of easily computing the scaling factor k that produces a desired similarity value.

3.3 The Weighted Power Reduction-Similarity-Product Metric

In Section 3.2 we have shown that the $MSSIM$ of a brightness-scaled image with respect to the input is roughly independent from pixel values. The same is not true for the power consumption of an OLED, as clear from (1): total power is proportional to the γ-th power of the total image luminance. Since brightness scaling reduces the luminance of each pixel of a factor k, the relation among the power of an image and that of its brightness scaled version is $P_{SCAL} \propto k^{\gamma} P$.

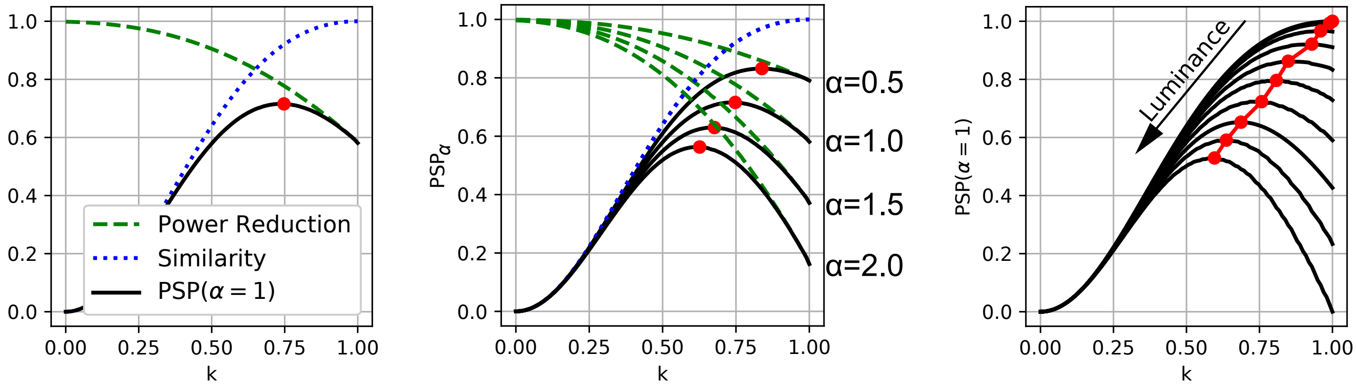
According to this equation, for a given k, applying brightness scaling to an image with high average luminance will yield more power benefits with respect to the same transformation on a dark image, due to the larger initial power P. At the same time, bright images are also those for which brightness scaling is more tolerable. In fact, although the numerical *similarity* ($MSSIM$) is invariant, dimming bright and dark images does not have the same effect on the perceived image *quality*. In the case of a bright image, brightness scaling makes it more dull and less beautiful, but does not prevent the intelligibility of its content. In contrast, scaling an already dark image might cause the loss of information content, depriving it from its usefulness. Such image-dependence of the power reduction and quality loss obtained by brightness scaling shows that an optimal dimming should adapt the scaling factor k to the features of the input image.

To quantify this intuition, [16] proposed a novel optimization metric, called *Power Reduction-Similarity Product* (PSP). The PSP is the product of two components, both functions of the scaling factor k. The first accounts for the power reduction obtainable through brightness scaling, the second for the corresponding effect on similarity. Both factors should be maximized, but their dependency on k is opposite, hence generating a trade-off.

However, the original PSP formulation in [16] implicitly gave the same "importance" to the power and similarity components. As a consequence, the optimal scaling factor was *always the same for a given image*, and there was no way for a user to configure the method in order to obtain different power saving/similarity trade-offs. In this work, we extend the original PSP function proposing a new *Weighted* PSP (or PSP_{α}), which allows to configure the desired *power reduction effort* through the new parameter α. The mathematical expression of the *Weighted* PSP is the following:

$$PSP(k, I)_{\alpha} = \left(1 - \frac{\alpha P(k, I)}{P_{MAX}}\right) MSSIM(I, kI) \tag{8}$$

where $P()$ is the total power consumption of the image and $MSSIM()$ is computed as described in Section 3.2. P_{MAX} is the maximum power that can be dissipated by the OLED panel, i.e. that of a totally white image. The similarity component of the PSP_{α} is simply the MSSIM between the original and scaled images, while the power component is the normalized power reduction obtained when displaying the scaled image. Through α, the weight of the latter term can be increased ($\alpha > 1$) or decreased ($\alpha < 1$), giving more or less importance to power reduction; $\alpha = 1$ yields the original PSP from [16].



(a) Example for a single image ($\alpha = 1$).

(b) Same example, for different values of α.

(c) Maximum point variation with respect to average luminance ($\alpha = 1$).

Fig. 2. Analysis of the proposed Weighted PSP metric.

Figure 2a shows the shapes of the two components of the PSP_α as a function of k for an example image, with α set to 1. The blue dotted curve is the approximate SSIM/MSSIM equation obtained in (7), while the green dashed curve represents the image-dependent normalized power reduction component, which decreases proportionally to $-k^\gamma$. Because of the opposite trends with respect to k, the PSP has a single maximum, highlighted with a red dot in the figure. The value of k corresponding to this point is the scaling factor that achieves the best balance between power reduction and similarity for that image: $k_{opt} = arg\max(PSP_\alpha(k, I))$.

Figure 2b shows how the maximum point of the PSP_α changes for different values of α. For example, when $\alpha = 2$ (i.e. double power reduction effort) the power curve becomes steeper compared to the case of $\alpha = 1$, consequently yielding a smaller k_{opt}. This is coherent, as smaller scaling factors should be selected when minimizing power consumption is more important than preserving the image. When $\alpha < 1$ (i.e. reduced power reduction effort) the behavior is the opposite.

Both these graphs still refer to a single example image. The plot in Figure 2c, instead, shows the variation of the maximum point of PSP_α for images of different luminance, in the case of $\alpha = 1$. The different curves have been obtained using monochrome gray images, with average luminance ranging from 0 (black) to 255 (white). When luminance increases, the power curve becomes steeper, and the optimum point moves to the left (smaller k_{opt}). This expresses the intuition that *the optimal scaling factor is smaller for brighter images*. For a fully black image obviously $k_{opt} = 1$; indeed, power consumption is already minimal and scaling has no effect. At the other extreme, i.e. for a white image, the optimal scaling factor converges to $k_{opt} \approx 0.59$.

Finally, Figure 3 shows the combined effect of the image-adaptiveness and of the power reduction effort parameter on the optimal scaling factor computed by LABS. Specifically, the graph shows the value of k_{opt} as a function of α and of the average image luminance, thus merging the effects expressed by Figures 2b and 2c.

4 Implementation Details

The main computation performed by LABS at runtime is the identification of the adaptive scaling factor k_{opt} for a target image. After k_{opt} has been determined, the rest of the transformation is a standard brightness scaling as in (2). Next, we describe two possible approaches to extract k_{opt}.

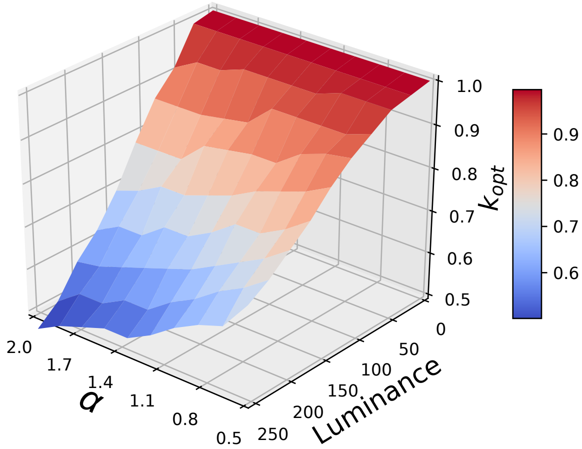


Fig. 3. Variation of k_{opt} with respect to average luminance and α.

4.1 Exact k_{opt} Computation

The ideal way to obtain k_{opt} is to compute the derivative of the PSP (8) with respect to k, and finding the value of k for which such derivative is equal to zero (i.e. its stationary point). In mathematical terms:

$$\frac{\delta PSP(k, I)_\alpha}{\delta k} = -4k \cdot \frac{A - B + C}{P_{MAX}(k^2 + 1)^3} = 0 \qquad (9)$$

where:

$$A = \alpha k(k^2 + 1)\frac{\delta P(k, I)}{\delta k} \quad B = 2\alpha(k^2 - 1)P(k, I)$$
$$C = 2P_{MAX}(k^2 - 1) \qquad (10)$$

$P(k, I)$ and P_{MAX} are defined as in (8), while $\delta P(k, I)/\delta k$ is the derivative of the power consumption equation (1) with respect to k. The latter can be computed combining (1) and (3b), obtaining:

$$\frac{\delta P(k,I)}{\delta k} = \sum_{i=0}^{W}\sum_{j=0}^{H}(\gamma d_{11} Y_{i,j} w_r \overline{R_{i,j}}^{\gamma-1} + \gamma d_{21} Y_{i,j} w_g \overline{G_{i,j}}^{\gamma-1} + \gamma d_{31} Y_{i,j} \overline{B_{i,j}}^{\gamma-1}) \quad (11)$$

In this equation, d_{11}, d_{21}, d_{31} are the first column elements of matrix $\mathbf{D}$ in (3b) and $\overline{R_{i,j}}$, $\overline{G_{i,j}}$, $\overline{B_{i,j}}$ are the RGB components of pixel (i,j) *after* scaling its luminance with k, which are therefore dependent on the scaling factor through (2) and (3b).

Equation (9) cannot be solved analytically in closed form, except for the trivial solution $k = 0$, which does not correspond to the desired PSP maximum. In fact, some of the terms include γ-th and $(\gamma - 1)$-th powers, which are in general non-integer. However, an "exact" solution up to an arbitrarily small tolerance can be obtained using numerical methods. Any method for finding the roots of a function can be adopted; in the following, we select Brent's method [30] due to its speed and convergence guarantees.

This numerical approach yields the exact k_{opt}, apart for the tolerance, but still requires a significant computational effort. Every iteration requires the computation the aforementioned non-integer powers, as well as divisions and summations over all pixels (several million for a HD image) included in $P(k,I)$ and $\delta P(k,I)$. Therefore, this approach is not suitable for usage in real-time, but it can still be applied to *static* images, e.g. in a "gallery" application for smartphones. Moreover, it is a fundamental starting point for the approximate method described hereafter.

4.2 Regression-based k_{opt} Approximation

For real-time adaptive scaling, we propose an alternative regression-based method to determine k_{opt}. This approach enables the computation of a close-to-optimal scaling factor for each frame being displayed, that can be obtained in software in very short time, or accelerated with simple hardware.

Starting from the observation of Section 3.3 that power, and hence k_{opt} are strongly affected by luminance, we use regression to correlate k_{opt} with the average luminance of the image Y_{avg}. This is clearly an approximation, since chrominance components also contribute to the total power, although with a smaller impact [3]. However, as shown later, the error with respect to the exact method is very low on average. Moreover, obtaining Y_{avg} at runtime has a low computational cost, and just consists of summing the $Y_{i,j}$ component of each pixel. The final division by $1/WH$ can be implemented as a constant multiplication. Moreover, the conversion to YCbCr, e.g. from RGB, is not an additional cost, as it is already part of most brightness scaling approaches.

To build the regression model, we perform an off-line training, in which Y_{avg} and the corresponding exact value of k_{opt} (using the numerical approach) are computed for a large set of representative images. These data are then used to determine the parameters of a *linear regression* model. As shown in Section 5, more complex models (i.e. higher order polynomials) do not yield a significant benefit in terms of estimation accuracy. In summary, the mathematical expression used to determine k_{opt} at runtime is as follows:

TABLE 1
Number of additions/subtractions and multiplications required by each phase of the online LABS transformation.

Phase	Add/Sub	Mul
RGB to YCbCr	$6WH$	$9WH$
Compute Y_{avg}	WH	1
k_{opt} Regression	1	1
Apply Eq. (2)	0	WH
YCbCr to RGB	$6WH$	$4WH$

$$k_{opt} \approx mY_{avg} + q \quad (12)$$

Evaluating this equation only involves additions and multiplications, besides the obvious control flow operations. Table 1 summarizes the number of operations required by each phase of the LABS *online* transformation, as a function of the image dimensions W and H. As shown, the overall complexity is linear in the number of pixels, i.e. $O(WH)$, and most operations are due to color-space conversions.

The operations involved in the regression-based method can be implemented with very low energy and time overheads, both in software and in hardware. While a software implementation is trivial, Figure 4 reports a high-level block diagram of a possible hardware implementation. The block labeled *Accumulator* is simply a flip-flop based register used to accumulate luminances into Y_{avg}, whereas *RGB to YCbCr* and *YCbCr to RGB* are circuits to perform color-space conversions, similar to those presented in [23]. Finally, notice that since both RGB and JPEG YCbCr formats require 24-bit, pixels can be overwritten in-place when changing the color space. Therefore, the hardware accelerator does not need any additional memory, besides the display frame buffer.

Notice that the regression-based approach still permits the configuration of the power reduction effort in LABS through the parameter α. This requires repeating the training phase for each value of α considered, and extracting the corresponding model coefficients (m_α, q_α). These coefficients can then be changed at runtime depending on the desired power reduction effort. In the hardware implementation of Figure 4, this simply implies that m and q must be stored in writable registers.

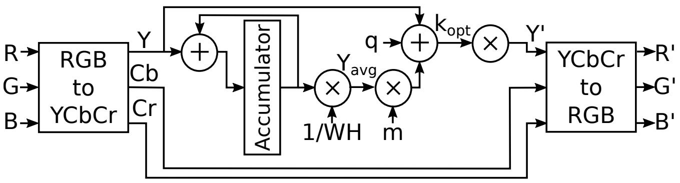
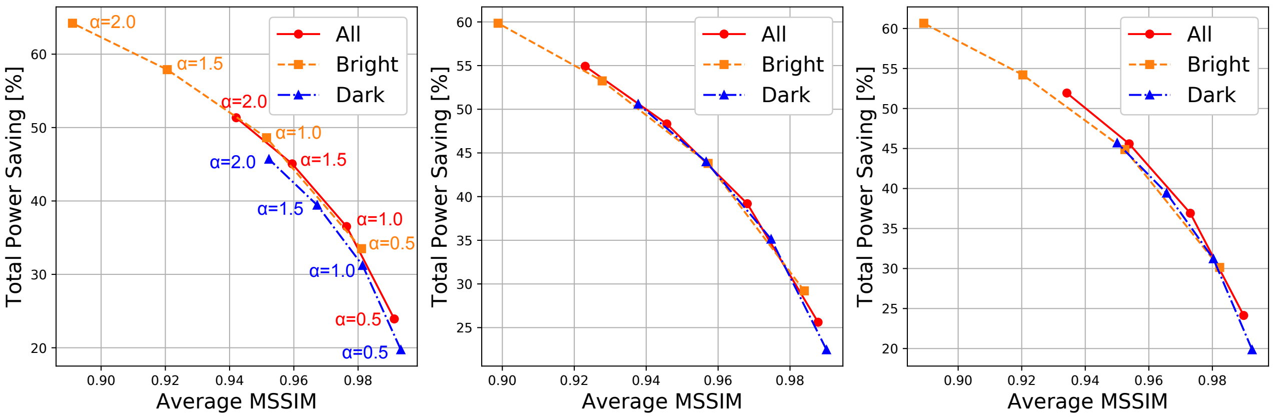


Fig. 4. Block diagram of a hardware accelerator for the online part of LABS, using the regression-based method for estimating k_{opt}.

5 Experimental Results

5.1 Setup

We have evaluated the performance of LABS on both still images and videos. We have used three publicly available



(a) Kodak (b) Inria (c) BSDS

Fig. 5. Power saving versus MSSIM Pareto curves for entire datasets and dark/bright subsets.

TABLE 2
Total power saving, average MSSIM, average PSNR and average k_{opt} for different datasets and values of α.

Data	Split	Tot. Power Saving [%]				Avg. MSSIM				Avg. PSNR				Avg. $\mathbf{k_{opt}}$			
α		0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0
Kodak	All	23.93	36.52	45.1	51.33	0.99	0.98	0.96	0.94	32.01	27.72	25.40	23.88	0.91	0.85	0.80	0.76
	Bright	33.5	48.61	57.89	64.21	0.98	0.95	0.92	0.89	25.28	21.34	19.33	18.07	0.85	0.77	0.71	0.66
	Dark	19.73	31.22	39.43	45.7	0.99	0.98	0.97	0.95	33.36	29.00	26.62	25.04	0.93	0.87	0.82	0.78
Inria	All	24.14	36.91	45.58	51.93	0.99	0.97	0.95	0.93	31.77	27.59	25.30	23.79	0.91	0.85	0.80	0.76
	Bright	30.13	44.87	54.20	60.65	0.98	0.95	0.92	0.89	26.09	22.04	19.96	18.65	0.86	0.78	0.72	0.67
	Dark	19.85	31.22	39.43	45.70	0.99	0.98	0.97	0.95	33.76	29.54	27.17	25.59	0.93	0.87	0.83	0.79
BSDS	All	25.62	39.19	48.32	54.92	0.99	0.97	0.95	0.92	29.67	25.49	23.26	21.81	0.89	0.82	0.76	0.72
	Bright	29.21	43.80	53.25	59.85	0.98	0.96	0.93	0.90	26.32	22.23	20.12	18.79	0.87	0.78	0.72	0.67
	Dark	22.47	35.13	43.99	50.60	0.99	0.97	0.96	0.94	31.71	27.47	25.17	23.65	0.91	0.84	0.79	0.75

image repositories: the *Kodak* dataset [28], the *INRIA Holiday dataset* [29], and the *Berkeley Segmentation Data Set* (BSDS) [27], each containing natural images with a variety of subjects and features (luminance, contrast, etc.). For videos, we have experimented on sequences from the *Open Video Project* [31] and from the *Derf's Test Media Collection* [32].

As a model of OLED power consumption we have used (1) with the same coefficients adopted in [10] and [16], i.e: $(w_0, w_r, w_g, w_b, \gamma) = (0, 70, 115, 154, 2.2)$.

We have used Python 3.5 and the scikit-learn library to implement the offline phase of the regression-based k_{opt} computation The online part of LABS (both the exact version based on Brent's method and the regression-based approach) has been implemented in C and compiled with LLVM 3.9.1. Software execution times have been measured on a Intel Core i7 processor running at 2.2GHz with 16GB of RAM. Although the typical target of LABS are ARM-based devices (smartphones, tablets, etc.), results on a desktop-class processor serve to demonstrate the speed-up achieved by our method compared to the state-of-the-art. The hardware version of the regression-based k_{opt} computation has been described in VHDL, and synthesized using Synopsys Design Compiler L-2016.03, for a 45nm standard cell library from STMicroelectronics. The clock frequency has been set to 1GHz. Execution time has been evaluated through gate-level simulations in Mentor QuestaSim 10.6, whereas power consumption has been estimated in Synopsys PrimeTime L-2016.06.

5.2 Power Saving versus MSSIM and Adaptivity

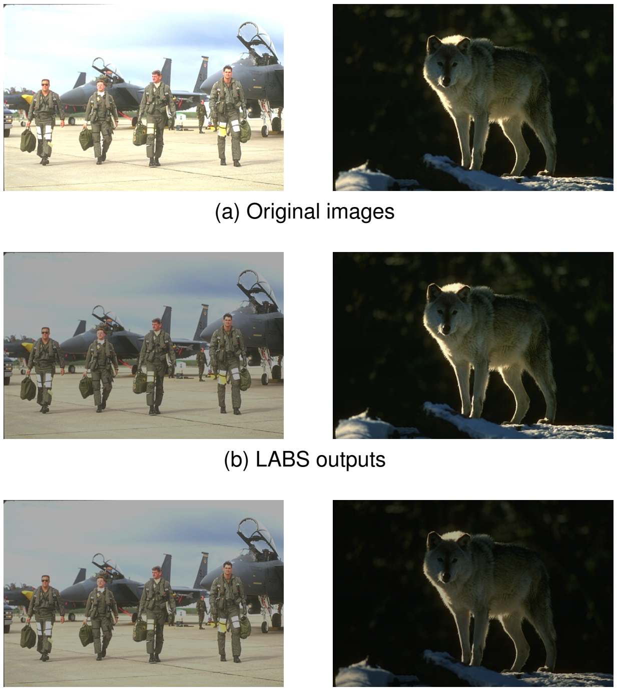
In this section, we demonstrate the image-adaptivity of LABS, and the possibility of exploring the power versus quality design space at runtime through the parameter α.

To show the effect of the power reduction effort parameter, we have computed the total power saving and average MSSIM obtained by our method on the three still images datasets, for different values of α, i.e. 0.5, 1.0, 1.5 and 2.0. Power saving has been computed in percentage, as if the OLED panel had been used to display each image of the dataset once, i.e.:

$$P_{SAV} = \left(1 - \frac{\sum_i P_{SCAL,i}}{\sum_i P_{ORIG,i}}\right) \cdot 100 \qquad (13)$$

where $P_{SCAL,i}$ is the power consumption for displaying the i-th image of the dataset after brightness scaling with k_{opt}, and $P_{ORIG,i}$ is the consumption of the same image without brightness scaling. Each dataset has been considered in its entirety (*All*), as well as after splitting it into two subsets: *Dark*, containing images with average luminance is $L < 0.5L_{MAX}$, and *Bright*, containing all the remaining images, where $L_{MAX} = 255$ in YCbCr.

The results of this analysis are reported in Figure 5 as Pareto curves. For clarity, one of the plots has been annotated with the corresponding value of α. As shown, LABS achieves similar and significant power savings on the three datasets, while maintaining the average MSSIM very close to 1. Specifically, with $\alpha = 1$ (the default setting), the



(a) Original images

(b) LABS outputs

(c) Uniform scaling outputs at iso-power

Fig. 6. Example of the advantages of LABS over uniform brightness scaling.

power reduction on the entire datasets is between 35% and 40%, and the MSSIM is always greater than 0.97. When the power reduction effort is doubled ($\alpha = 2$) the total savings become $> 50\%$, at the cost of a relatively small drop in similarity (MSSIM ≈ 0.93).

The adaptivity of our method is shown by the fact that savings and similarity values change when considering *Dark* or *Bright* images separately. Bright images are scaled more aggressively, hence they yield larger savings for a given value of α. Detailed results of this experiment are reported in Table 2. The table also reports the average PSNR (in dB) between original and transformed images, and the average value of k_{opt} determined by LABS for different values of α. As expected, the average k_{opt} is smaller for brighter images and decreases for larger α. On the other hand, its value remains approximately constant across datasets.

A visual example of the advantages of our adaptive approach is shown in Figure 6. Given the strongly different luminance among the two original images (Figure 6a), it is clear that the leftmost one has a significantly larger power consumption than the rightmost one. Specifically, with the model of (1), the brighter image contributes to $\approx 95\%$ of the total power for displaying both of them in sequence on a OLED display. When transforming the two images using LABS, our algorithm determines a value of $k_{opt,l} \approx 0.68$ for the leftmost image, and $k_{opt,r} \approx 0.96$ for the rightmost one. LABS outputs are shown in Figure 6b. The bright image has become less bright, to reduce power consumption as much as possible; however, its content is still well preserved. On the contrary, the dark image is virtually unchanged, since the power saving obtainable by scaling it more aggressively would not be beneficial, compared to the corresponding loss in visibility. Globally, the total OLED power for displaying both images is reduced of 55.1%.

Figure 6c shows the results of a standard (image-independent) brightness scaling as a comparison. For fairness, we have determined a *single* (uniform) scaling factor that, when applied to both images, provides the *same total savings* obtained with LABS (55.1%) . Such scaling factor has been determined numerically, through a bisection method, and the resulting value is $\tilde{k} \approx 0.70$. As expected, this value is similar to $k_{opt,l}$, as the leftmost image contributes to most of the total power. However, the visual output of uniform scaling is significantly worse than that of LABS; while there are no visible benefits on the brighter image, the darker one is significantly less clear, and many of its details are lost.

5.3 Application to Video Sequences

The advantages of LABS are even more evident when applied to video sequences. We have processed some of the sequences from [31] and [32] with LABS, setting $\alpha = 1$; Figure 7 shows the frame-by-frame scaling factors produced by LABS for the first 2000 frames of one of these videos (*Elephants Dream*). The plot reports also some key frames, corresponding to relevant changes of k_{opt}. Notice how the scaling factor follows the content of the video, becoming smaller when the frame brightness increases and vice versa.

A side-by-side comparison of the original and scaled videos for comparison is available at the URL in [33]. From these videos it can be seen that, although k_{opt} is recomputed at every new frame, this does not generate any flickering artifacts. In fact, abrupt changes in k_{opt} occur only in correspondence of scene changes, where the brightness of the frame changes significantly. Within a single scene, k_{opt} varies only slightly, thus not generating visible disturbances.

5.4 Regression Analysis

In order to reduce the overheads of LABS, in Section 4 we have proposed to use linear regression to put in relation the average image luminance Y_{avg} and k_{opt}. In this section, we verify the effectiveness of a regression-based approach.

For this experiment, we have used the BSDS dataset [27], which is already conveniently split into training and test subsets. First, we have computed k_{opt} for both training and test images using the exact numerical method, and annotated the corresponding Y_{avg}. Then, we have used the training set to determine the parameters of several *polynomial regression* models, changing the model order from 1 (linear) to 4. Finally, the estimated k_{opt} obtained with each model on test images have been scored. The experiment has been repeated with two different values of α.

The numerical results of this analysis are reported in Table 3, which shows the Root Mean Squared Error (RMSE) of each model, the maximum error over the entire test set, and the R^2 score. Figure 8 shows the test images on a Y_{avg} versus k_{opt} plane, together with the various models outputs, for the case $\alpha = 1$. As shown in Table 3, the linear regression model yields very low errors compared to the exact k_{opt} computation: the RMSE values are around 0.019 for $alpha = 1$ and 0.026 for $alpha = 2$, and the R^2 values are around 0.88 independent of α. The worst case difference between the exact and estimated k_{opt} is around 0.05 ($\alpha = 1$)

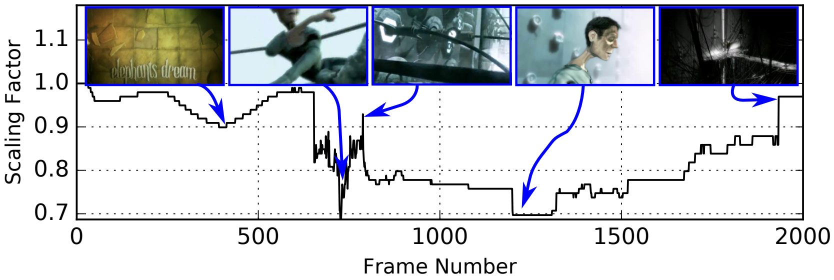


Fig. 7. Sequence of scaling factors for the first 2000 frames of the *"Elephants Dream"* video from [32].

and 0.075 ($alpha = 2$), which are barely distinguishable for humans.

The most important observation from the table, however, is that using higher-order models does not provide significant benefits and rather yields a worse average errors.

TABLE 3
RMSE, maximum error and R^2 score on the BSDS test set, for different polynomial regression models.

	$\alpha = 1$			$\alpha = 2$		
Order	**RMSE**	**Max Err.**	$\mathbf{R^2}$	**RMSE**	**Max Err.**	$\mathbf{R^2}$
1	0.0188	0.0535	0.880	0.0261	0.0744	0.884
2	0.0189	0.0540	0.878	0.0268	0.0778	0.870
3	0.0189	0.0526	0.880	0.0267	0.0797	0.871
4	0.0190	0.0528	0.878	0.0268	0.0796	0.870

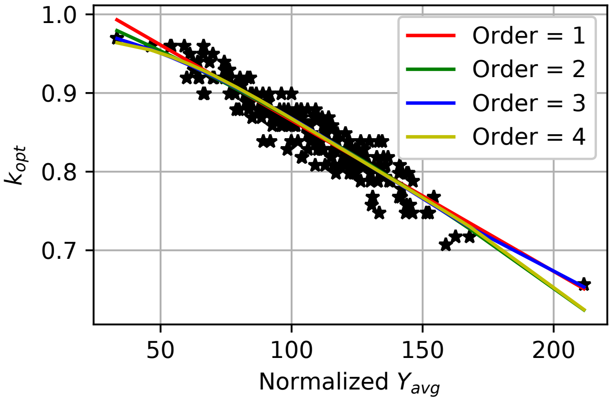


Fig. 8. Polynomial regression of k_{opt} as a function of Y_{avg} for the BSDS dataset, with $\alpha = 1$. Linear model slope: $m = -0.34$, bias: $q = 1.05$.

5.5 Software Implementation Results

Table 4 reports a detailed breakdown of the software execution time of LABS, split into its main sections, for two different image resolutions. Since the number of operations involved in LABS is data independent, these results are valid for any image of the same size. The k_{opt} *Computation* section can be performed either with the *Exact* method or with the *Regression*-based method, as explained in Section 4, hence the corresponding column reports both results. For each section, the table reports the average execution time and the standard deviation over 1000 runs.

For comparison, the last column reports the execution time of the *Perceptual Quality-aware Power Reduction* (PQPR) technique proposed in [24]. Notice that PQPR uses exactly the same luminance transformation of LABS and only differs in the computation of the adaptive scaling factor.

When considering the exact method based on Brent's algorithm, LABS is $2.5x$ faster than PQPR for a 512x512 image, and $3.9x$ for a 1920x1080 image, thus denoting also a better scaling. However, the exact solution is still too slow for a real-time implementation: execution times are in the order of hundreds of milliseconds, about $10x$ more than the display refresh rate.

This motivates the use of the regression-based approach, which is $48x$ faster than the exact method for the low-res images and $30x$ for the high-res ones. The speedups with respect to PQPR are $121x$ and $117x$, respectively. Notice that the execution time is dominated by color-space conversions: additional operations to implement adaptive brightness scaling only require 0.38 ms and 3.34 ms respectively.

In summary, assuming a refresh rate of 15 ms, the software version of LABS based on regression can be used in real-time for low-res images (e.g. 512x512). However, for high-res the time required by color-space conversion motivates the need for hardware acceleration.

5.6 Hardware Implementation Results

The main metrics of the building blocks of a hardware accelerator implementing LABS are reported in Table 5. We do not report any state-of-the-art result for comparison, since to the best of our knowledge, LABS is the first adaptive brightness scaling approach for which a hardware implementation is proposed.

In hardware, some of the sections of our algorithm are implemented using pipelining: as soon as a pixel has been converted to YCbCr, its luminance is added to the *Accumulator* register (see Figure 4). Similarly, a pixel is converted back to RGB immediately after its luminance has been scaled. This is the reason why the latencies of the first two and of the last two sections are "grouped" in the table. Energy values are computed assuming that each block only consumes when used, and is shut off (e.g. by power/clock gating) when idle. The accelerator uses fixed-point for all intermediate coefficients (m_α, q_α, k_{opt}, etc.), and data bitwidths are selected to ensure an average error on the output pixels smaller than 0.5% with respect to software, which uses double precision floating point.

The hardware implementation of LABS is significantly faster than the software one, despite its small silicon area and low power consumption. The entire transformation (including color-space conversion) takes about 4ms for a

TABLE 4
Software execution time of LABS (in milliseconds) and comparison with PQPR [24]

Image Size	Method	RGB to YCbCr	k_{opt} Computation	Luminance Scaling	YCbCr to RGB	LABS (Total)	PQPR [24]
512 x 512	**Exact** **Regression**	1.64 ± 0.15	180.39 ± 6.99 0.12 ± 0.01	0.26 ± 0.05	1.84 ± 0.15	184 ± 6.99 **3.86 ± 0.22**	468 ± ?
1920 x 1080	**Exact** **Regression**	13.21 ± 1.04	917.89 ± 33.25 1.05 ± 0.31	2.29 ± 0.57	15.03 ± 1.13	948 ± 33.29 **31.58 ± 1.67**	3700 ± ?

TABLE 5
Detailed metrics of the LABS hardware implementation.

Section	Area [mm^2]	Power [mW]	Latency/ Frame [ms]		Energy/ Frame [μJ]	
			512x 512	1920x 1080	512x 512	1920x 1080
RGB-YCbCr	0.013	0.98	0.26	2.07	0.26	2.04
k_{opt} Comp.	0.020	0.61			0.16	1.26
Lum. Scaling	0.005	0.33	0.26	2.07	0.09	0.67
YCbCr-RGB	0.008	0.53			0.13	1.09
LABS (Total)	0.046	2.45	0.52	4.14	0.64	5.06

high-res image and can then be used in real-time. Most importantly, the *energy* overheads for implementing the transformation are negligible compared to the reductions obtained on the display. In fact, let us take a small (240x320 pixels) AMOLED panel as a conservative example [34] with a typical power consumption of 260 mW. A refresh rate of 15 ms corresponds to an energy consumption per frame of $0.26W \cdot 0.015ms = 3.9$ mJ, about 500x more energy than that of the accelerator for a 512x512 image (Table 5). Therefore, the power savings reported of Table 2 can be considered as true savings.

Notice also that all hardware results in Table 5 refer to pixel-serial implementations, that process one pixel per clock cycle. However, parts of the architecture of Figure 4 would easily lend themselves to parallelization, allowing to explore the execution time versus cost design space.

6 Conclusions

We have presented LABS, a low-overhead adaptive brightness scaling transformation for OLED displays. By optimizing the tradeoff among power reduction and image similarity, balanced through a tunable parameter, LABS automatically determines an optimal image-dependent scaling factor. Despite achieving significant power reductions on a wide variety of images (more than 35% for less than 3% drop in MSSIM), LABS yields itself to a low-complexity implementation, both in software and in hardware. This allows its application at runtime and in real-time, with negligible energy overheads.

We foresee this technique being used in mobile consumer devices (smartphones, tablets, etc.) as an energy-driven yet easy to use display configuration.

References

[1] J. N. Bardsley, "International OLED technology roadmap," *IEEE J. Sel. Topics Quantum Electron.*, vol. 10, n. 1, pp. 3–9, jan 2004.

[2] V. C. Bender, T. B. Marchesan, and J. M. Alonso, "Solid-State Lighting: A Concise Review of the State of the Art on LED and OLED Modeling," *IEEE Ind. Electron. Mag.*, vol. 9, n. 2, pp. 6–16, jun 2015.

[3] M. Dong and L. Zhong, "Chameleon: A Color-Adaptive Web Browser for Mobile OLED Displays," *IEEE Trans. Mobile Comput.*, vol. 11, n. 5, pp. 724–738, may 2012.

[4] M. Dong and L. Zhong, "Power Modeling and Optimization for OLED Displays," *IEEE Trans. Mobile Comput.*, vol. 11, n. 9, pp. 1587–1599, sep 2012.

[5] P. Ranganathan et al., "Energy-aware user interfaces and energy-adaptive displays," *IEEE Computer*, vol. 39, n. 3, pp. 31–38, mar 2006.

[6] C.-H. Lin, C.-K. Kang, and P.-C. Hsiu, "Catch Your Attention," in *Proc. DAC '14*. 2014, pp. 1–6.

[7] M. Linares-Vásquez et al., "Multi-objective optimization of energy consumption of GUIs in android apps," *ACM Trans. on Software Engineering and Methodology (TOSEM)*, vol. 27, no. 3, p. 14, 2018

[8] D. Shin et al., "Dynamic Driver Supply Voltage Scaling for Organic Light Emitting Diode Displays," *IEEE Trans. on Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, n. 7, pp. 1017–1030, jul 2013.

[9] X. Chen et al., "Fine-grained dynamic voltage scaling on OLED display," in *Proc. 17th ASP-DAC*. 2012, pp. 807–812.

[10] C. Lee et al., "Power-Constrained Contrast Enhancement for Emissive Displays Based on Histogram Equalization," *IEEE Trans. Image Process.*, vol. 21, n. 1, pp. 80–93, jan 2012.

[11] N. Yeon-Oh et al., "Power-Constrained Contrast Enhancement Algorithm Using Multiscale Retinex for OLED Display," *IEEE Trans. Image Process.*, vol. 23, n. 8, pp. 3308–3320, aug 2014.

[12] C. Y. Jang et al., "Noniterative Power-Constrained Contrast Enhancement Algorithm for OLED Display," *J. Display Technol.*, vol. 12, n. 11, pp. 1257–1267, nov 2016.

[13] Y.-T. Peng et al., "Histogram shrinking for power-saving contrast enhancement," in *Proc. IEEE ICIP*. 2013, pp. 891–894.

[14] C.-K. Kang et al., "A win-win camera: Quality-enhanced power-saving images on mobile OLED displays," in *Proc. IEEE/ACM ISLPED*. 2015, pp. 267–272.

[15] X. Chen et al., "MORPh," in *Proc. DAC '16*. 2016, pp. 1–6.

[16] D. Jahier Pagliari et al., "Optimal content-dependent dynamic brightness scaling for oled displays," in *Proc. 27th PATMOS*. 2017, pp. 1–6.

[17] Z. Wang et al., "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, n. 4, pp. 600–612, 2004.

[18] A. K. Bhowmik and R. J. Brennan, "System-Level Display Power Reduction Technologies for Portable Computing and Communications Devices," in *Proc. IEEE Portable*. 2007, pp. 1–5.

[19] Wei-Chung Cheng and M. Pedram, "Power minimization in a backlit TFT-LCD display by concurrent brightness and contrast scaling," *IEEE Trans. Consum. Electron.*, vol. 50, n. 1, pp. 25–32, feb 2004.

[20] C.-C. Lai and C.-C. Tsai, "Backlight power reduction and image contrast enhancement using adaptive dimming for global backlight applications," *IEEE Trans. Consum. Electron.*, vol. 54, n. 2, pp. 669–674, may 2008.

[21] C. Y. Jang et al., "Perceived Distortion-Based Progressive LCD Backlight Dimming Method," *J. Display Technol.*, vol. 12, n. 10, pp. 1130–1138, oct 2016.

[22] J. Shin and R.-H. Park, "Power-constrained contrast enhancement for organic light-emitting diode display using locality-preserving histogram equalisation," *IET Image Processing*, vol. 10, n. 7, pp. 542–551, jul 2016.

[23] D. Jahier Pagliari et al., "LAPSE: Low-Overhead Adaptive Power Saving and Contrast Enhancement for OLEDs," *IEEE Trans. Image Process.*, vol. 27, n. 9, pp. 4623–4637, 2018.

[24] S.-j. Kang, "Perceptual Quality-Aware Power Reduction Technique for Organic Light Emitting Diodes," *J. Display Technol.*, vol. 12, n. 6, pp. 519–525, jun 2016.

[25] S.-J. Kang, "Image-Quality-Based Power Control Technique for Organic Light Emitting Diode Displays," *J. Display Technol.*, vol. 11, n. 1, pp. 104–109, jan 2015.

[26] *Information technology - Digital compression and coding of continuous-tone still images; JPEG File Interchange Format* , ITU Recommendation T.871, May 2011.

[27] P. Arbelaez et al., "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, n. 5, pp. 898–916, May 2011.

[28] "Kodak image database." http://r0k.us/graphics/kodak/

[29] H. Jegou et al., "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. ECCV'08*. 2008, pp. 304–317.

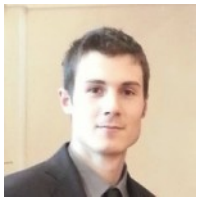
[30] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *Comput. J.*, vol. 14, n. 4, pp. 422–425, 1971.

[31] "The open video project." Online: open-video.org

[32] "Derf's test media collection." Online: https://media.xiph.org/video/derf/

[33] [Online]. URL: https://vimeo.com/album/4457406

[34] "USMP-A24240TP AMOLED Product Specification.", Ver 1.2, US Micro Products, 2008.



Daniele Jahier Pagliari (M15) received the M.Sc. and Ph.D. degrees in Computer Engineering from Politecnico di Torino, Italy, in 2014 and 2018 re- spectively. He is currently Assistant Professor at the same institution. His main research interests focus on computer-aided design of digital systems, with particular emphasis on low-power optimization and approximate computing.

Santa Di Cataldo (M07) is Assistant Professor (tenure track) in Politecnico di Torino, Italy since 2017. She received her Biomedical Engineering degree from Politecnico di Torino in 2006. She holds a Ph.D. in Systems and Computer Engineering from the same university since April 2011. In the same year she joined the Department of Control and Computer Engineering in Politecnico di Torino as a Post-Doctoral researcher. Her main research interests are image processing, machine learning and heterogeneous data integration, including techniques for pattern recognition, image segmentation, feature quantification and classification.

Edoardo Patti (M16) is Assistant Professor at Politecnico di Torino. He received both M.Sc. and Ph.D. degrees in Computer Engineering at Politecnico di Torino in 2010 and 2014, respectively. During the Academic Year 2014/2015, he was Academic Visiting at the University of Manchester. His research interests concern: i) Ubiquitous Computing; ii) Internet of Things; iii) Smart Systems and Cities; iv) Software architectures with particular emphasis on infrastructure for Ambient Intelligence; v) Software solutions for simulating and optimising energy systems; vi) Software solutions for energy data visualisation to increase user awareness. In the fields above, he has authored over 70 scientific publications (between 2011 and 2018).

Alberto Macii is a Full Professor of Computer Engineering at Politecnico di Torino, Torino, Italy. He holds a Laurea Degree in Computer Engineering from Politecnico di Torino and a Ph.D. degree in Computer Engineering from the same institution. His research interests are in the design of electronic digital circuits and systems, with particular emphasis on low-power consumption aspects. In the field above he has authored around 200 scientific publications. He received the Best Paper Award for an article presented at ACM/IEEE GLS-VLSI-08. He was an Associate Editor for the IEEE Transactions on Computer-Aided Design on Circuits and Systems and the Design Contest Co-Chair of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED). Alberto Macii is Senior Member of the IEEE.

Enrico Macii (SM02, F07) is a Full Professor of Computer Engineering at Politecnico di Torino, Italy. Prior to that, he was an Associate Professor (1998-2001) and an Assistant Professor (1993-1998) at the same institution. From 1991 to 1997 he was also an Adjunct Faculty at the University of Colorado at Boulder. He holds a Laurea Degree in Electrical Engineering from Politecnico di Torino (1990), a Laurea Degree in Computer Science from Universit di Torino (1991) and a PhD degree in Computer Engineering from Politecnico di Torino (1995). From 2009 to 2016, he was the Vice Rector for Research at Politecnico di Torino; he was also the Rector's Delegate for Technology Transfer (2009-2015) and for International Affairs (2012-2015). His research interests are in the design of electronic circuits and systems, with particular emphasis on low-power consumption, optimization, testing, and formal verification. In the last few years, he has been growingly involved in projects focusing on the development of new technologies and methodologies for smart cities and bioinformatics. In the fields above he has authored over 450 scientific publications. Enrico Macii is a Fellow of the IEEE.

Massimo Poncino (SM12, F18) received the Ph.D. degree in Computer Engineering and the Dr. Eng. degree in Electrical Engineering from the Politecnico di Torino, Italy. He is currently a Full Professor of Computer Engineering at Politecnico di Torino. His research interests include several aspects of design automation of digital systems, with particular emphasis on the modeling and optimization of low-power systems. He is the author or coauthor of more than 350 journal and conference papers. He is an Associate Editor of the ACM Transactions on Design Automation of Electronic Systems and of IEEE Design & Test. Prior to that, he was an Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2006-2012). He was the Technical Program Co-Chair (in 2011) and the General Chair (in 2012) of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED). He serves on the Technical Program Committee of several IEEE and ACM technical conferences, including DAC, ICCAD, DATE, ISLPED, ASP-DAC, CODES-ISSS, and GLSVLSI. Massimo Poncino is a Fellow of the IEEE.