Optimal Content-Dependent Dynamic Brightness Scaling for OLED Displays

Optimal Content-Dependent Dynamic Brightness Scaling for OLED Displays

Daniele Jahier Pagliari, Enrico Macii, Massimo Poncino
Dipartimento di Automatica e Informatica
Politecnico di Torino, Turin, Italy
Email: {daniele.jahier, enrico.macii, massimo.poncino}@polito.it

***Abstract*—Brightness scaling is the most common way to reduce power consumption in OLED displays. Such "dimming" is generally static, i.e. it is applied either manually by the user, or automatically by the system in correspondence of predefined battery state-of-charge conditions. This is obviously sub-optimal, because it makes brightness adaptation (i) too coarse-grain in time, and (ii) agnostic of the image being displayed.**
In this work, we overcome the two above limitations by proposing a novel brightness scaling approach based on the online calculation of a *content-dependent optimal dimming factor*. While keeping the simplicity of traditional scaling, this solution enables a true dynamic scaling applicable on a frame-by-frame basis.
Results show that the proposed strategy is able to obtain an average power saving greater than 30% for different reference image datasets, while maintaining the Mean Structural Similarity Index (MSSIM) between the original and transformed images at about 97%. We also perform an analysis of the costs for both hardware and software implementations of the transformation.

I. Introduction

Organic Light Emitting Diode (OLED) displays are increasingly used in various types of electronic devices, thanks to their superior features (thinner, brighter, better viewing angles, etc.) with respect to LCDs [1]. Since displays often account for a large share of system power [2], OLED consumption must be carefully managed. The emissive nature of OLEDs makes their power strongly *image-dependent*; while they are generally more efficient than traditional LCDs, they consume significantly more for bright images [1].

This need for energy efficiency has spurred a wide body of methods for reducing power in OLEDs, that almost invariably act on the intrinsic tradeoff between "quality" of the image and achievable saving. The most explored solution is based on applying an *image transformation* to the display content. Since OLED power is determined by pixel intensities, transforming the image can significantly reduce consumption [3], [5]–[7]. In practice, most transformations consist of pixel intensity mappings, i.e. application of a function $f(x)$ to each pixel x of the image. The various method differ in the shape of $f()$, and in the possibility of having $f()$ defined *statically* or *dynamically*, i.e. changing at runtime depending on the image. Among the virtually infinite image transformations, the simplest is $f(x) = kx$, in which each pixel intensity is scaled by a quantity $k < 1$. This is the typical behavior of the "brightness" knob in mobile 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).

While simple and effective, this standard *brightness scaling* has two main drawbacks. Firstly, it is virtually **static**, since the factor k is controlled manually by the user, and only occasionally by the system. Secondly, it is totally **agnostic of the content being displayed**. On the contrary, it would be desirable to have the amount of scaling *depend on the power consumption of the displayed image*: images that consume more (i.e. brighter ones) should be scaled more with respect to darker ones. As an extreme case, a black image could not be scaled at all, since the achievable saving would be zero.
In this work we propose a new brightness scaling approach that tackles the two above issues. In particular:

- We propose to *optimize, for a given image, the tradeoff between power saving and image alteration*, using a combined power/similarity metric. This allows tuning the amount of brightness scaling according to the image content, making it more "aggressive" for images with larger power consumption.
- We show that the expression of this metric can be computed analytically in closed form, and has *a single optimum per image*.
- Finally, we show that the optimum scaling factor identified by the metric can be put in relation with easy-to-compute image features (total luminance) by means of regression. This allows building a simple yet accurate expression to compute the optimum at runtime, with small software/hardware overheads.

Results show that the proposed strategy is able to obtain an average power saving superior to 30% for different reference image datasets, with virtually no loss in similarity (about 97% on average using MSSIM [8]). We also analyze the costs of hardware and software implementations of the proposed approach, thus demonstrating its applicability at runtime.

II. Background and Related Work

OLED displays pixels consist of three *emissive* devices, one for each RGB channel [2]. The intensities of color components depend on the current flowing through each device. Therefore, the total consumption of the panel is strongly *affected by the displayed image*. Measurements on real OLED panels allowed to build an empirical power consumption model [2] :

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

In (1), 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 [2]. Normally, $\gamma \in [2:3]$.

Power optimization techniques for OLEDs are tailored to suit this new image-dependent power model. They can be broadly split in two categories: those targeting Graphical User Interfaces (GUIs), and those applicable to general images.

Techniques for GUIs drastically change colors [2] or selectively dim parts of the image that are outside of the user focus area [9]. Both methods are based on the observation that, for GUIs, *usability* is more important than visual *fidelity*.

When targeting general images (e.g. pictures and videos) instead, fidelity becomes the main objective. Thus, most techniques try to homogeneously alter the image *luminance*, leaving chromatic components untouched, as the latter strongly undermine fidelity [3]–[7], [10].

The solution in [10] applies Dynamic Voltage Scaling (DVS) to OLED panel drivers, hence reducing the current flow through the emissive devices. This produces a luminance reduction, which is then compensated acting on pixel values. Despite its effectiveness, this technique requires modifications of the panel electronics (insertion of custom drivers and control circuits), and cannot be applied to off-the-shelf panels.

Several alternatives have been proposed that save power *solely* transforming the displayed image. Power Constrained Contrast Enhancement (PCCE) [3] is an image-adaptive luminance transformation that concurrently reduces power and enhances contrast. A variant of PCCE based on multiscale retinex is described in [4]. In [5] a similar transformation is proposed, that combines Histogram Shrinking with contrast enhancement. In all these solutions, identifying and applying the transformation requires complex iterative procedures (e.g., non-linear optimization), which cannot be realistically implemented at runtime for each new displayed frame. In fact, modern displays refresh rates are in the order of $50 - 60Hz$, hence the entire process has to be repeated every $\approx 15 - 20ms$. The only way to reach this throughput for the methods in [3]–[5] is to employ powerful hardware, e.g. a processor core solely dedicated to transforming display frames, but this would result in very large energy and cost overheads, and significantly reduce transformation savings.

For applications at runtime, the most common image transformation for OLED power reduction is brightness scaling [6], [7]. For grayscale images, brightness scaling transforms pixels with the function: $x' = k \cdot x$, where x is the pixel value and k is called *scaling factor*. For color images, the same effect is obtained scaling the luminance component of a pixel. To do so, the pixel is first converted to an appropriate color space, such as YCbCr, where the Y component corresponds to luminance. The conversion from RGB to YCbCr and vice versa is obtained through linear transformations with constant coefficients [11]. Despite its simplicity, brightness scaling is very effective. Indeed, (1) implies $P \propto Y^{\gamma}$, hence scaling luminance has a strong effect on power consumption [3]. However, the traditional implementation of brightness scaling is *image-agnostic*, i.e. it uses a *single* scaling factor for all images. Moreover, the selection of the scaling factor is driven either by the user, acting on a setting knob, or by the operating system, reacting to some external event, such as a change in the ambient illumination or a decrease in the battery state-of-charge. In both cases, the scaling factor is modified with very coarse temporal granularity (e.g., from seconds to hours).

In [7], the authors propose a fine-grain image-dependent scaling solution, based on the search of a k that minimizes power, under image alteration (MSSIM [8]) constraints. However, their formulation requires very complex calculations, that make it virtually impossible to implement at runtime (about 2 seconds of execution for a $1920x1080$ image).

III. Image-Dependent Brightness Scaling

A. Motivation and Goal

In this paper, we propose a new image-dependent brightness scaling method that lends itself to a simple implementation, thus allowing real-time adaptivity with very low overheads. Our method concurrently optimizes two contrasting metrics: display power reduction and image similarity. By doing so, it obtains an *optimal scaling factor for each displayed image*. This is different from previous solutions, in which of the two metrics was optimized using the other as a constraint [6], [7]. Our solution can be leveraged to replace the traditional brightness control knob in mobile devices with a single *on/off button*, that activates adaptive brightness scaling when desired. This would allow to automatically optimize the visibility and power consumption of each image, and relieve the user from the task of manually setting the desired brightness level.

B. MSSIM and Brightness Scaling

The starting point of our approach is the analysis of the relation between brightness scaling and image similarity. The Structural SIMilarity (SSIM) index [8] is a popular metric that well correlates with how humans perceive image similarity:

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

In (2), x and y are subsets of the image pixels, obtained via a sliding window; μ_x, μ_y and σ_x^2, σ_y^2 are the mean and variance of pixels gray levels in x and y, and σ_{xy} 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, $\mu_x = \mu_y = 0$). In the case of color images, gray level is replaced with luminance [11]. SSIM values for all positions of the sliding window are then averaged to compute the global Mean SSIM (MSSIM).

As noted in [7], if pixels in y are luminance-scaled versions of those in x, i.e. $y = kx$, then: $\mu_y = k\mu_x$, $\sigma_y = k\sigma_x$, $\sigma_{xy} = k\sigma_x^2$. Consequently (2) can be rewritten as:

$$SSIM(x,kx) = \frac{(2k\mu_x^2 + c_1)(2k\sigma_x^2 + c_2)}{[(1+k^2)\mu_x^2 + c_1][(1+k^2)\sigma_x^2 + c_2]} \quad (3)$$

The authors of [7] did not consider however that, for the vast majority of images, (3) is approximately *independent* from the

pixel values. In fact, since c_1 and c_2 are only introduced for stabilization, their value is chosen to have negligible impact on the metric. As a matter of fact, standard values 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 [12].
It can be seen easily that, with the pixel statistics of typical images, the error introduced by removing these two constants from (3) is very small. Even for small scaling factors, the terms $2k\mu_x^2$ and $2k\sigma_x^2$ are one or two orders of magnitude greater than c_1. Removing c_1 and c_2 from (3), we get the following approximate expression:

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

Therefore, for a luminance scaling transformation, the SSIM (and thus the MSSIM) *approximately depends only on the scaling factor, and not on the values of image pixels*. This is experimentally confirmed by the box plots in Figure 1, which show the distribution of MSSIM values between original and luminance-scaled ($k = 0.8$) images, calculated over three different image datasets [13]–[15]. As shown, most of the images have MSSIM values differing by less than 0.5%.

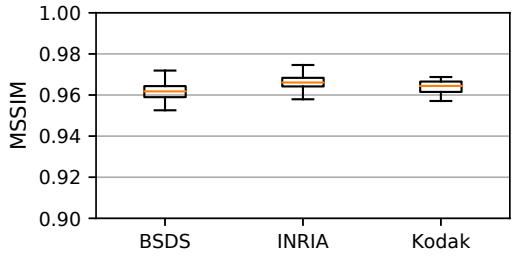


Fig. 1: MSSIM distribution for three different image datasets after luminance scaling with $k = 0.8$.

This apparently counter-intuitive finding makes sense when considering that the SSIM/MSSIM is a metric of *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.

C. The Power Reduction-Similarity-Product (PSP) Metric

If the MSSIM of a brightness-scaled image is roughly independent from the pixel values, the same is not true for power consumption. As mentioned in Section II, the 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 power of a scaled image is $P_{SCAL} \propto k^\gamma P$. Hence, applying a "stronger" brightness scaling (smaller k) on high-luminance images provides greater power benefits, as the initial power P is larger. Incidentally, brighter 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 perceived *quality*: while the content of a dimmed bright image is most likely still intelligible, scaling an already dark image might totally deprive it of its information content.

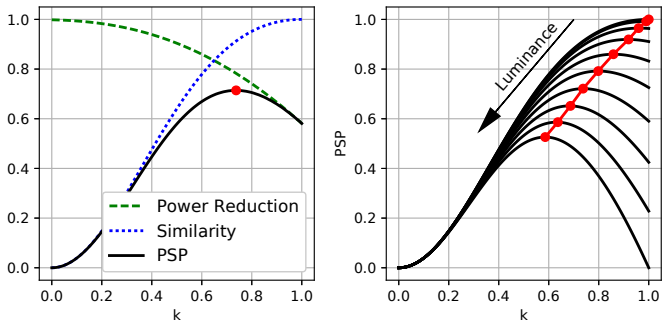


Fig. 2: Analysis of the PSP metric. Example for a single image (left) and maximum point variation with respect to the image luminance (right).

Therefore, we can identify *a relation between the original image and the optimal scaling factor k to be applied*, in order to minimize power while preserving as much as possible the information conveyed by the image.
To quantify this intuition, we propose a novel optimization metric, which we call *Power Reduction-Similarity-Product* (PSP). As the name says, 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 individually maximized, but their dependency on k is opposite, hence the trade-off. The mathematical expression for the PSP is the following:

$$PSP(k, I) = \left(1 - \frac{P(k, I)}{P_{MAX}}\right) MSSIM(I, kI) \tag{5}$$

where $P()$ is the total power consumption of the image and $MSSIM()$ is computed as described in Section III-B. 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 is simply the MSSIM of the original and scaled images. The power component, instead, represents the (normalized) power reduction obtained when displaying the scaled image. Normalization ensures that both components vary in the range $[0 : 1]$ for any image.
The shapes of the two components and of the PSP as a function of k are shown in the left side of Figure 2 for an example image. The blue dotted curve is the approximate SSIM/MSSIM equation obtained in (4), while the green dashed curve represents the image-dependent 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: $k_{opt} = arg\max(PSP(k, I))$.
The right side of Figure 2 shows how the maxima of PSP change for images with different total luminance. The curves refer to monochrome images, with average luminance values ranging from 0 (black) to 255 (white).

As luminance increases, the power reduction curve becomes more steep; for instance, for a totally white image ($P(I) = P_{MAX}$), the dashed green curve of Figure 2 reaches 0 for $k = 1$. Consequently, k_{opt} decreases as luminance increases, expressing the intuition that 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 $k_{opt} \approx 0.59$.

D. Runtime Adaptive Scaling Implementation

Obtaining k_{opt} numerically requires computing the derivative of (5) with respect to k, and finding the value of k for which $\delta PSP(k, I)/\delta k = 0$. For runtime adaptive scaling, k_{opt} must be recomputed for each new display frame, with minimal energy overheads. Thus, it must be either obtained in software, in very short time, or accelerated with simple hardware.

However, even using the approximate MSSIM expression of (4), the PSP derivative still includes divisions, as well as γ-th and $(\gamma - 1)$-th roots, computed for every pixel. Since γ is generally not an integer, these operations must be implemented with numerical methods [16]. Doing so at runtime, under the constraints determined by the display refresh rate, would either consume most of the main system CPU time just for transforming images, or require an additional dedicated processor, with huge energy overheads.

To overcome this problem, we devise a low-overhead correlation-based approach to approximate k_{opt} at runtime. Specifically, we correlate k_{opt} to the *total luminance* of the image (Y_{tot}), in accordance to the observation of Section III-C that power, and hence k_{opt}, are strongly affected by luminance. Computing Y_{tot} at runtime is straightforward, and just requires summing the Y component of each pixel in YCbCr space[1].

To model this correlation, we perform an initial off-line *training phase*, in which Y_{tot} and the corresponding *exact* value of k_{opt} are computed using (5) for a large set of representative images. These data are then used to determine the parameters of a regression equation. Clearly, this method produces an *approximation* of the real k_{opt}, since pixel *chroma* components also contribute to power (and hence affect k_{opt}), although with a smaller impact compared to luminance [2]. Moreover, small image-dependent MSSIM variations are also neglected. However, as shown in Section IV, even a simple *linear* regression model achieves very good accuracy. Thus, we can approximate k_{opt} as:

$$k_{opt} = mY_{tot} + q \tag{6}$$

Computing k_{opt} at runtime reduces to evaluating this equation, at the cost of one multiplication and one addition with constant coefficients. These operations have very low energy and time overheads, both in software and in hardware, allowing to adapt the scaling factor at runtime in a very efficient way. A block diagram of the operations involved in the adaptive brightness scaling transformation is reported in Figure 3.

[1]The conversion to YCbCr, e.g. from RGB, is not an additional cost, as it is already part of most brightness scaling approaches.

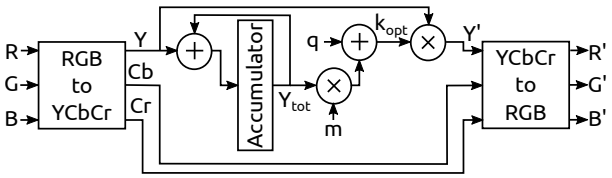


Fig. 3: Block diagram of the operations involved in the proposed adaptive brightness scaling technique.

IV. Experimental Results

A. Experimental Setup

To assess the performance of the proposed solution, we tested it on three publicly available image datasets: the classic *Kodak* dataset (24 images) [14], the *INRIA Holiday dataset*, (800+ images) [15], and the *Berkeley Segmentation Data Set* (BSDS), (500 images) [13]. All three contain images with a wide variety of subjects, luminance and contrast. For experiments on videos, we used sequences from the *Open Video Project* [17] and from the *Derf's Test Media Collection* [18]. For the estimation of OLED power consumption we adopted the model of (1), with the same numerical coefficients used in [3], i.e. $(w_0, w_r, w_g, w_b, \gamma) = (0, 70, 115, 154, 2.2)$.

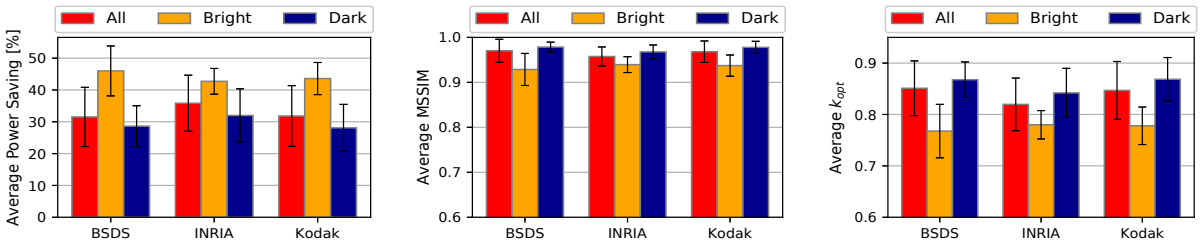
We used Python 3.5 to perform training and compute regression coefficients. The software version of the online transformation was written in C and compiled with LLVM 3.9.1. Execution time was measured on a 2.2GHz Intel Core i7 processor, with 16GB of RAM. The hardware version was designed in VHDL and synthesized using Synopsys Design Compiler L-2016.03, for a 45nm standard cell library from ST Microelectronics. The clock frequency was set to 1GHz. Execution time was evaluated through gate-level simulations in Mentor QuestaSim 10.6, whereas power consumption was estimated in Synopsys PrimeTime L-2016.06.

B. Average Power Saving and Adaptivity

As a first experiment, we computed the average power saving obtained by our approach on the three datasets. We also analyzed the corresponding average MSSIM and the optimal scaling factor k_{opt}. Power saving was computed in percentage with respect to the original (not scaled) image, i.e. $P_{SAV} = \left(1 - \frac{P_{SCAL}}{P_{ORIG}}\right) \cdot 100$. Additionally, to better show the adaptivity of our technique, we repeated the evaluation after splitting each dataset in two subsets, according to luminance. We denoted as *Dark* the images whose total luminance is $L < 0.5L_{MAX}$, and as *Bright* the remaining ones, where L_{MAX} is the luminance of an equally sized white image. Results are shown in Figure 4; black segments indicate the intervals defined by the samples standard deviations.

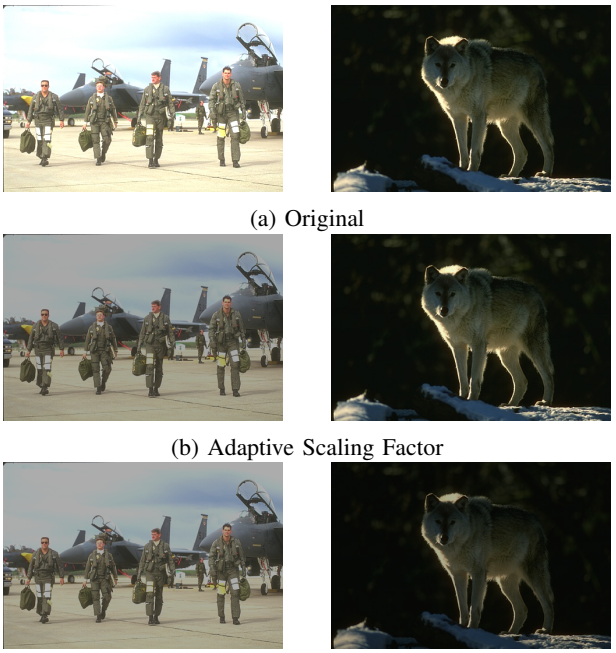
As shown in the plots, the average saving is $> 30\%$ for all three datasets, with quite stable variability among images. As expected, the technique saves more power on bright images, using a smaller k_{opt}. While this has an impact on the MSSIM, the average index is always > 0.97 for entire datasets, and never lower than 0.92 even for bright images.

A visual example of the benefits of our adaptive approach is shown in Figure 5. According to the model of (1), the



(a) Power saving (b) MSSIM (c) Optimal scaling factor (k_{opt})

Fig. 4: Average power saving, MSSIM and k_{opt}, before and after splitting into dark and bright subsets.



(a) Original

(b) Adaptive Scaling Factor

(c) Uniform Scaling Factor

Fig. 5: Example of the advantages of the proposed approach.

power consumption for displaying the two images of Figure 5a on an OLED panel is mostly determined by the leftmost, brighter one, ($\approx 95\%$ of the total), while the rightmost only contributes for $\approx 5\%$. When applying the proposed technique, computations yield $k_{opt,l} \approx 0.68$ for the leftmost image, and $k_{opt,r} \approx 0.96$ for the rightmost one. The results of applying these scaling factors are shown in Figure 5b. The leftmost image has become less bright, to reduce power consumption as much as possible; however, its content is well preserved. On the contrary, the dark image is practically unchanged, since the saving obtainable by scaling it more aggressively is not beneficial, compared to the corresponding loss in visibility. Globally, the saving obtained for the two images is 55.1%.

For sake of comparison, the results of a standard, image-agnostic brightness scaling at iso-power are shown in Figure 5c. To obtain these images, we computed (numerically, with a bisection method) a *single* scaling factor that, when applied to both images, provides the same total saving as before (55.1%). The result is $\tilde{k} \approx 0.70$. Predictably, the value is similar to $k_{opt,l}$, as the leftmost image accounts for most of the total power. However, applying the same scaling factor to both images is clearly inferior to our approach; while there

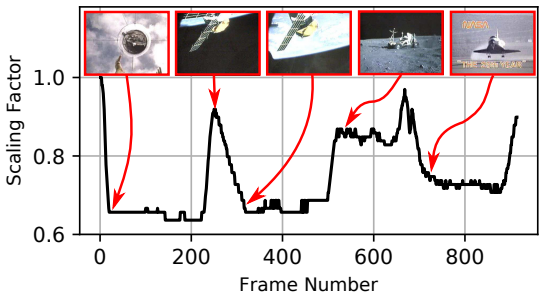


Fig. 6: Sequence of scaling factors for the *"Nasa Anniversary 01"* sequence from [17].

are no visible benefits on the brighter image, the darker one is significantly duller, and many details are lost.

C. Application to Video Sequences

The advantages of the proposed approach are more evident in videos. We processed several sequences from [17] and [18] with adaptive brightness scaling, and put them side by side with originals for visual comparison. Results have been uploaded to the URL in [19]. Figure 6 shows the sequence of scaling factors obtained for one of these sequences, called *Nasa Anniversary 01* [17]. Some key frames have been highlighted to better follow the variation of k_{opt} in time. Notice how our algorithm follows the content of the video, scaling more aggressively when brightness increases, and vice versa. The sequences in [19] also show that, although the scaling factor is recomputed at every new frame, this does not generate any flickering in the videos. In fact, abrupt changes in k_{opt} only occur in correspondence of scene changes, in which the brightness of the frame changes significantly. Within a single scene, k_{opt} remains approximately constant, thus not generating visible artifacts.

D. Regression Analysis

Our method uses regression for building a simplified relation between total image luminance Y_{tot} and the optimal scaling factor k_{opt}. We tested this solution on the BSDS dataset [13], which is already split into training and test images. To this end, we computed Y_{tot} and k_{opt} for both training and test sets. We used the training set to determine the slope m and the bias q of the linear regression model presented in Section III-D, which we validated against the test set. Model output and test data are shown in Figure 7, in which the total luminance has been normalized to a $[0:1]$ range for visualization.

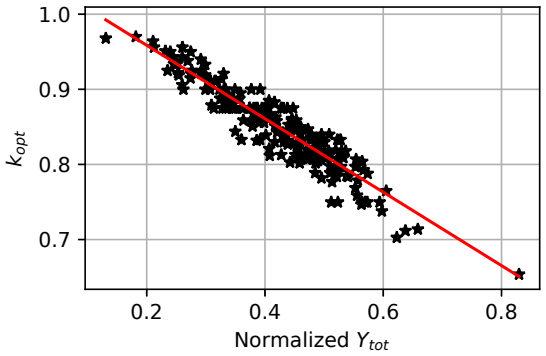


Fig. 7: Linear regression of k_{opt} as a function of Y_{tot} for the BSDS database [13]. Model slope: $m = -0.34$, bias: $q = 1.05$. Scores: $R^2 = 0.89$, $RMSE = 0.01$, $MAXE = 0.05$.

The good R^2 value confirms the goodness of the model. More importantly, the Root Mean Squared Error ($RMSE$) and the Maximum Error ($MAXE$), show that the difference between exact and predicted k_{opt} values is on average smaller than 0.01 (a value completely indistinguishable for the human eye) and never larger than 0.05. These errors are mainly due to neglecting the effect of chroma on power. However, their limited magnitude shows that the proposed model achieves a very good compromise between complexity and accuracy.

E. Evaluation of Implementation Overheads

Results of software and hardware implementations of the proposed adaptive scaling technique, under the conditions described in Section IV-A are shown in Table I for two different image sizes. Since all operations involved in the transformation have data-independent complexity, these results are valid for *any* image of the given size. The execution time of the software version has been averaged over 1000 runs.

Software execution time does not include color-space transformations phases (RGB to YCbCr and vice versa), in order to fairly compare our approach with PCCE [3]. For a 512x512 image, our method requires less than $0.5ms$, against the $6.23ms$ reported in [3] for the same size. Although our method does not enhance images as PCCE does, it achieves a speedup of more than 10x, while still maintaining image adaptivity. Thus, it has the remarkable advantage of being applicable at runtime, even for large panel sizes.

Hardware results in Table I, instead, refer to the full transformation, including color-space conversions. The hardware uses fixed-point representations for all intermediate coefficients (m, q, k_{opt}, etc.), and the data bit-widths 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 very small energy and area costs, and the short execution times, confirm the convenience of the proposed technique.

For sake of comparison, the datasheet of a 240x320 pixels AMOLED panel reports a typical power consumption of 260 mW [20]. With a refresh rate of 15 ms, this corresponds to an energy consumption per frame of $0.26 \cdot 0.015 = 3.9 \cdot 10^{-3} J$. Despite the fact that the panel is even *smaller* than the minimum image considered in Table I, the energy consumption

	SW	HW				
Image Size	**Time [ms]**	**Net Count**	**Area [mm^2]**	**Power [mW]**	**Time [ms]**	**Energy/ Frame [μJ]**
512x512	0.49	2557	0.0058	2.89	0.52	1.50
1920x1080	4.59				4.14	11.97

TABLE I: Software and CMOS implementation results.

of the display is three orders of magnitude higher compared to that of the additional hardware implementing the transformation. Thus, the theoretical savings computed in Section IV-B will be almost exactly preserved in a real system.

V. Conclusions

We have presented a new adaptive brightness scaling solution for OLED displays. By maximizing the combination of power reduction and image similarity, this method automatically selects a scaling factor that adapts to the displayed image. On average, the proposed solution achieves more than 30% power saving on a wide variety of images. Moreover, we have demonstrated a simple way to implement it at runtime with low software or hardware overheads. This technique could be implemented in mobile devices, providing the user with a single energy saving on/off button, and relieving him/her from the task of manually setting a display brightness level.

References

[1] J. Bardsley, "International oled technology roadmap," *IEEE J. Sel. Topics in Quantum Electron.*, vol. 10, no. 1, pp. 3–9, Jan 2004.

[2] M. Dong and L. Zhong, "Power modeling and optimization for oled displays," *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1587–1599, Sept 2012.

[3] C. Lee et al, "Power-constrained contrast enhancement for emissive displays based on histogram equalization," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 80–93, Jan 2012.

[4] Y.-O. Nam, D.-Y. Choi, and B. C. Song, "Power-constrained contrast enhancement algorithm using multiscale retinex for oled display," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3308–3320, Aug 2014.

[5] Y.-T. Peng et al, "Histogram shrinking for power-saving contrast enhancement," in *20th IEEE ICIP*, 2013, pp. 891–894.

[6] S. J. Kang, "Image-quality-based power control technique for organic light emitting diode displays," *IEEE J. Display Technol.*, vol. 11, no. 1, pp. 104–109, Jan 2015.

[7] S. J. Kang, "Perceptual quality-aware power reduction technique for organic light emitting diodes," *IEEE J. Display Technol.*, vol. 12, no. 6, pp. 519–525, June 2016.

[8] Z. Wang et al, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, April 2004.

[9] P. Ranganathan et al, "Energy-aware user interfaces and energy-adaptive displays," *Computer*, vol. 39, no. 3, pp. 31–38, March 2006.

[10] D. Shin et al, "Dynamic voltage scaling of oled displays," in *48th ACM/EDAC/IEEE DAC*, 2011, pp. 53–58.

[11] A. Koschan and M. Abidi, *Digital Color Image Processing*. Wiley, 2008.

[12] ITU-T, Recommendation T.871, May 2011.

[13] P. Arbelaez et al, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

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

[15] H. Jegou et al, "Hamming embedding and weak geometric consistency for large scale image search," in *10th ECCV*, 2008, pp. 304–317.

[16] I. Koren, *Computer Arithmetic Algorithms*. Wiley, 2008.

[17] G. Geisler and G. Marchionini, "The Open Video Project: Research-oriented Digital Video Repository," *5th ACM Conf. on Digital Libraries*, 2000, pp. 258–259.

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

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

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