Power-Saving Color Transformation of Mobile Graphical User Interfaces on OLED-based Displays

Mian Dong Yung-Seok Kevin Choi Lin Zhong
Department of Electrical & Computer Engineering, Rice University, Houston, TX 77025
{dongmian, ykc1,lzhong}@rice.edu

ABSTRACT

Emerging organic light-emitting diode (OLED)-based displays have drastically different power consumption when displaying different colors, due to their emissive nature. They bring a new opportunity for power saving by transforming GUI colors. In this work, we study this opportunity using a commercial-off-the-shelf QVGA OLED module and user studies. We present techniques that adapt GUIs based on existing mechanisms as well as arbitrarily under usability constraints. Our measurement and user studies show that more than 75% display power reduction can be achieved with user acceptance.

Categories and Subject Descriptors

D.2.2 [**Design Tools and Techniques**]: User Interfaces

General Terms

Algorithms, Measurement, Human Factors

Keywords

OLED Display, Graphic User Interface, Low Power

1. INTRODUCTION

Displays have been known as one of the major system energy consumers in mobile systems. In conventional liquid crystal displays (LCD), the adjustable external lighting dominates the power consumption; and the LCD panel consumes almost constant power regardless of the display content or GUI design [1].

In contrast, emerging organic light-emitting diode (OLED)-based displays [2] bring a new opportunity in power savings because their pixels are light-emitting and the power contribution made by a pixel is determined by the pixel's color. Henceforth, the power consumption by the OLED-based display varies significantly, depending on the display content. As a result, the GUI design will have a huge impact on the energy cost of the mobile applications. There is a great need for tools to automatically transform GUIs for power reduction. Firstly, most existing GUIs are designed for conventional LCDs without power consideration. Automatic transformation can readily adapt them for power savings on OLED-based displays. Moreover, transformation tools can help GUI designers effectively explore the large design space when power is considered.

The objective of this work is to provide such a tool, its theoretical foundation and evaluation. Our methodology is to transform the colors of a given GUI to minimize the power while satisfying given perceptual constrains. We first demonstrate that power-saving color transformation can be structurally applied to GUI themes and background/foreground colors. We then present a method for unstructured transformation, which applies to GUIs given in either image or code specification. The method first gathers statistics of pixel numbers of each color and then identifies a new color for each original color so that the color-transformed GUI has reduced power consumption and the difference between new colors remains in certain level of that between their originals. We evaluate the proposed transformations with both user studies and measurement based on a commercial-off-the-shelf QVGA OLED module. We show that the automatic transformation achieves over 75% power reduction with user acceptance.

To the best of our knowledge, this is the first public study that addresses GUI color transformation for power saving on OLED-based displays. The transformation methods presented here provide mechanisms for mobile operating systems and applications to construct energy-conserving policies for OLED-based display systems. They also enable mobile GUI designers to build adaptable and energy-efficient user interfaces. While many have investigated the power optimization of conventional LCDs, these works do not leverage the extraordinary flexibility provided by OLED-based displays. A recent few works did leverage such flexibility [3-5] but their exploitation was limited to display darkening [3, 5] and color inversion [4].

It is important to note that the proposed color transformation only applies to GUIs where *usability*, instead of *fidelity*, matters to the end users. Therefore, it does not apply to display of video or images. From this perspective, the proposed power-saving color transformation is complementary to existing LCD-based solutions that transform screen content under fidelity constraints [6-8].

The rest of the paper is organized as follows. We present background and address related work in Section 2. We describe our experiment setup in Section 3. We discuss structured color transformation based on existing color mechanisms in modern mobile platforms in Section 4. We propose a methodology for unstructured color transformation in Section 5. We address the limitation of this work and conclude in Section 6.

2. RELATED WORK

Researchers from HP Labs proposed to selectively darken part of the screen while maintaining good usability for OLED-based mobile displays [3, 5, 9]. IBM researchers sought to use low-power colors for more pixels and designed GUI objects, such as fonts and icons to minimize the need for high-power colors, on their Linux Wrist Watch [10]. The work, however, only studied the GUIs with two colors, i.e., background and foreground, without addressing colorful designs. There is a large body of work on energy optimization of conventional LCD systems. It reduces external lighting and compensates the change by transforming the displayed content [6, 7]. In particular, the characteristics of human visual perception were leveraged in [8]. Most of these techniques can be applied to OLED-based displays by scaling the luminance

ISLPED'09, August 19-21, 2009, San Francisco, California, USA.

level of OLED pixels. Their focus on preserving the content fidelity limits their effectiveness, although it makes them applicable to all display content. In contrast, we focus on GUIs for which *usability*, instead of *fidelity*, is the concern.

3. EXPERIMENTAL SETUP

We recruited twenty subjects from Rice University for user evaluation in this work. Their ages range between 22 and 27. All but four were males. None of the subjects had used an OLED-based display before. Because of the number of subjects and the way that they were recruited, we do not make any claim that they represent any particular demography.

For user evaluation, we use a 2.8" OLED QVGA display module from 4D Systems [11] to show color transformed GUIs to the subjects. For automatic color transformation, one needs an OLED display power model that estimates the power consumption, given the display content. Therefore, we leverage a pixel-based power model of 99% accuracy from our prior work on OLED display power modeling [12].

4. STRUCTURED TRANSFORMATION

Modern GUIs are usually structured. A GUI is composed of multiple objects, each with properties, e.g. size, location, and color, specified in software. In most cases, most pixels of a GUI object have one of three colors, i.e. border color, background color and foreground (usually text) color. In addition, most platforms support color themes in which GUI objects enjoy structurally consistent color patterns. Such structured provide an opportunity for color transformation without jeopardizing usability.

4.1 Background/Foreground Transformation

The background color usually claims most of the pixels of a GUI. Unfortunately, white, the most power-hungry color, is commonly used as the background color. Therefore, the most straightforward power-saving transformation is to employ a low-power color for background but colors of high contrast for border and foreground.

The research question is whether doing so will jeopardize user satisfaction of the GUIs. To answer this question, we design 15 very simple background/foreground schemes using black, grey, and white as background colors, and use black, white, red, green, and blue as foreground colors. We apply these schemes to a GUI that only includes texts, load them to the OLED module described in Section 3, and show them to our subjects one by one.

We ask our subjects to choose their top 3 favorite from the 15 background/foreground color schemes without knowing their power consumption. The three most popular schemes are grey/white, white/, and black/green, with 18, 16, 15 votes, respectively. With similar user satisfaction, however, the power consumption of the three schemes are dramatically different, i.e., 1.2 W, 3.0 W, 0.7 W, respectively. This demonstrates that *it is possible to achieve both low-power and high user satisfaction in background/foreground transformation.*

4.2 Theme-based Transformation

Modern mobile platforms usually support color themes that apply consistent colors to platform-specific features, e.g. title bar and borders. The themes therefore provide another structure to transform GUI colors for power saving.

To study the power impact of themes, we choose eight different themes from the Windows Mobile Themes website [13], each with over 10 review scores from the website. We apply them to eight sample programs with 10 GUIs from the Windows Mobile 5.0 SDK R2. We measure the power consumption of the 10 GUIs with each theme. Figure 1 shows the power consumption and average user review scores of the eight themes. As shown in the Figure 1, there is no obvious relation between power consumption and review scores. This indicates that *proper theme designs can achieve both low-power and high user acceptance.*

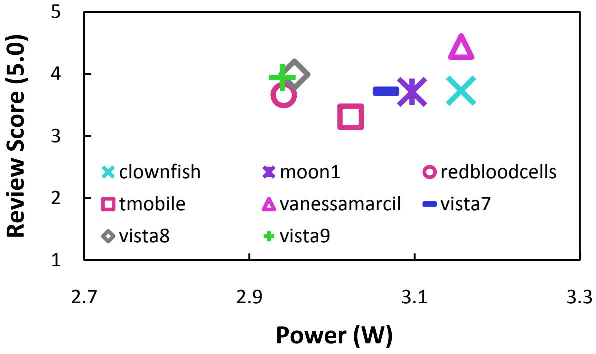


Figure 1. Power consumption and user review scores of GUIs with different themes

5. UNSTRUCTURED TRANSFORMATION

We present a much more flexible unstructured transformation method that can be applied to any GUIs. We first count the number of pixels for each color in a GUI to get a color histogram. Then, we map each of the original colors to a new one based for lower power consumption.

5.1 Color Counting

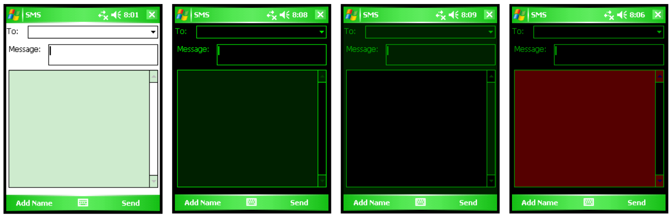
Using the power models described in Section 3, we are able to calculate the power consumption of a single pixel of any color. To get the power consumption of the whole display, we must count how many pixels for each color in the GUI, or a pixel number histogram of all colors. For a GUI specified by image, we can obtain the histogram by simply enumerating all the pixels. We can also apply sampling method to tradeoff between accuracy and computing cost. For a GUI specified by code, we use the following algorithm to obtain the color histogram. Suppose the list *Obj* and *ColorList* record all the objects and colors in the GUI, respectively. We go through all the objects in *Obj* and check whether its colors, including background, foreground and border colors, is in the *ColorList*, and add in the colors that are not yet in the list.

5.2 Color Mapping

After a color histogram of a GUI is obtained, we seek to reduce the power consumption of the OLED-based display by replacing each color of the histogram with a new color.

5.2.1 Problem Formulation

We formulate this process as a minimization problem. A GUI employs multiple colors, $c_1, c_2, \dots, c_n$, each of which is specified by a three component vector, either in sRGB space or CIELAB space, i.e., $c_i = (R_i, G_i, B_i) = (L_i^*, a_i^*, b_i^*), i = 1,2,\dots,n$. We can count how many pixels color c_i has and denote the number as num_i, which we obtain from color accounting. Then the power consumption of the GUI can be calculated as $P_{GUI} =$



Original **Green (75%)** **Green (50%)** **Color64 (100%)**

Figure 2. Unstructured transformed GUIs with different settings

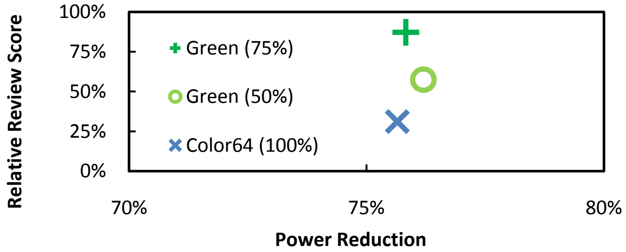


Figure 3. Power reduction vs. relative review score

$\sum_{i=1}^{n} num_i \cdot P_{pixel}(R_i, G_i, B_i)$, where $P_{pixel}(R,G,B)$ is the power model described in Section 3. The object of color mapping is to find n colors, $c_1', c_2', \ldots, c_n'$, such that power consumption $\sum_{i=1}^{n} num_i \cdot P_{pixel}(R_i', G_i', B_i')$ is minimized, while satisfying certain human perception constraints.

Color difference, defined as the Euclidean distance in CIELAB color space (L^*, a^*, b^*), is a widely used metric to measure human perceived difference [14] between colors. Two colors are considered perceptually identical when the difference between the two is less than certain threshold. Note previous work attempted to transform the screen while preserving its fidelity , i.e., there should be little human-perceptible difference between the transformed screen and the original [8]. In our case, however, we do not preserve the fidelity of GUI images. Instead, we only keep the usability, i.e., human users still perceive different colors in the transformed GUI as different as their originals. Thus, we ensure that the color difference between every two colors in the transformed GUI is no less than their counterparts in the original.

Thus, we formulate the problem as

$$\min \quad \sum_{i=1}^{n} num_i \cdot P_{pixel}(R_i', G_i', B_i')$$

$$\text{s.t.} \quad \forall i, j \in \{1,2,\ldots n\}$$

$$\left(L_i^{*\prime} - L_j^{*\prime}\right)^2 + \left(a_i^{*\prime} - a_j^{*\prime}\right)^2 + \left(b_i^{*\prime} - b_j^{*\prime}\right)^2$$

$$\geq \alpha \cdot \left(\left(L_i^* - L_j^*\right)^2 + \left(a_i^* - a_j^*\right)^2 + \left(b_i^* - b_j^*\right)^2\right);$$

$$0 \leq R_i' \leq 1;\ 0 \leq G_i' \leq 1;\ 0 \leq B_i' \leq 1.$$

Since it is likely that no solution exists that satisfies all the constraints, we employ a scalar α to indicate how strictly we want to enforce the perception constraints.

5.2.2 Rank-based Mapping

The most straightforward solution is *brutal-force search*, in which we divide the each of the R, G, and B component into discrete intervals and traverse all the possible combinations to find the one with minimal power consumption. Obviously, brutal-force search guarantees to find the optimal solution, but its computation time becomes unacceptably long when n is large. The computation complexity is $O(n^2(m_R m_G m_B)^n)$, where m_R, m_G, and m_B are the intensity levels of R, G, and B component, respectively.

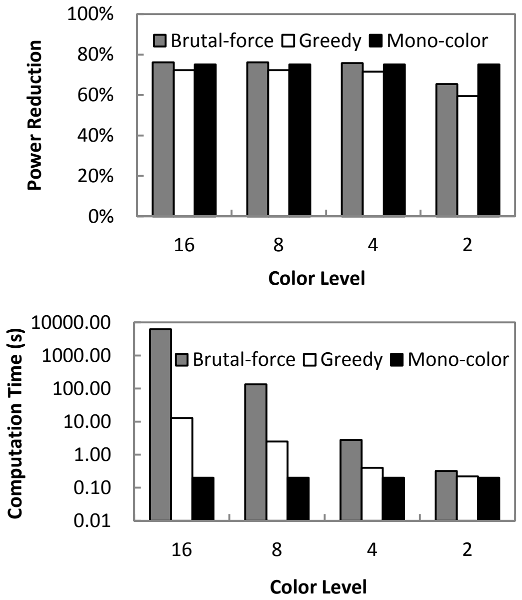


Figure 4. Power reduction and computation time with various color level

We can use a *greedy* algorithm to speed up the search process, in which it determines the colors one by one and seeks to achieve minimal power consumption at each step. It first chooses the color with zero power, i.e., black, to transform the color with the highest pixel number. And then it determines the remaining colors based on the rank of pixel number obtained in color counting. For each color, it seeks to achieve the minimal power consumption of the color itself while satisfying the constraints. The computation complexity is $O(n^3 m_R m_G m_B)$.

5.2.3 Mono-color Mapping

By default, the color mapping can select from all colors. We can also add constraints so that only a subset of the color space can be used. For example, by keeping $a_i^{*\prime}$ and $b_i^{*\prime}$ constant, we are able to perform a mono-color mapping, in which all the new colors share the same chromatic property but are with different luminance. Apparently, we can still use rank-based mapping to solve the problem. In this case, however, the constraints become linear and the objective function regarding L^* is convex, which guarantees that we can find solutions in polynomial time, e.g. the interior-point method. Moreover, as we see later, mono-color mapping provides higher user acceptance than multi-color mapping.

5.3 EVALUATION

5.3.1 Experiment Results

We implement the unstructured color transformation on .NET Compact Framework and C#, for Windows Mobile-based mobile embedded systems. We use the same 10 GUIs described in Section 4.2 to evaluate the implementation. As originals, we set the 10 GUIs in the vista9 (green) theme which has the lowest power consumption (See Figure 1). For each GUI, we perform transformations with three different settings.

- **Green (75%):** We perform a mono-color mapping only with color green as used in the theme, and we set α as 75% since there is no solution when α is 100%.
- **Green (50%):** To evaluate the impact of α, we perform another mono-color mapping with α as 50%.
- **Color64 (100%):** We also perform a multi-color mapping, in which each of the R, G, and B components is divided into four intensity levels resulting 64 different colors in total. In this case, we set α as 100%.

One original and three automatically generated GUIs are shown in Figure 2. Figure 3 shows the average power reduction of the 10 GUIs generated using the three transformations. As shown in the figure, each transformation achieves over 75% power reduction. Optimization effort α does not make much difference in power reduction although it results huge difference in their appearances.

As described in Section 5.2, intensity level has a huge impact on the computation time of the rank-based mapping algorithm. We perform experiments to study the trade-off between computation time and power reduction by changing the number of intensity level in color space (See Figure 4). We investigate four different cases with the number of the intensity level of each color equals to 16, 8, 4, and 2. As shown in the figure, using a coarse granularity, or small intensity level, will dramatically reduce computation time while achieving comparable power reduction. The reason is that every benchmark in the experiment has a white background that contains most pixels of the GUI and the majority of power reduction is achieved by mapping the white background into black.

5.3.2 User Evaluation

We used the same apparatus setup and procedure to perform user evaluation for the four color schemes. We apply these schemes to the 10 GUIs described in Section 4.2, load them to the OLED module described in Section 3, show them to our subjects one by one, and ask them to score them according to their fondness in 1 to 5 scale (5 means “like the most”). During the study, the subjects were not aware the power of the schemes.

Since the original theme gets the highest score by all the 20 subjects, we use it as the baseline to normalize the review scores of other themes. Figure 3 presents the evaluation results. As shown in the figure, mono-color mapping is more acceptable to users than multi-color mapping. Between the two mono-color transformed GUI color schemes, the one that enforces tighter perception constraints gets higher scores. Therefore we note: *(1) mono-color mapping rather is preferred over multi-color mapping; (2) Tighter perception constraints may lead to higher user acceptance with comparable power reduction*

Furthermore, after the study, we asked the subjects whether they prefer the low-power schemes over the originals, given the 75% power reduction in the display.18 out of 20 subjects said Yes.

6. CONCLUSIONS

In this work, we present a methodology of GUI optimization for saving power on OLED-based displays color transformation. We have the following findings. (1) Using appropriate background/foreground color schemes may save a lot of power without sacrificing user satisfaction. For example, black/green is a low-power scheme with high user acceptance. (2) Color transformation is able to reduce power by as much as 75% with little degradation in user satisfaction. And our user study shows that most of participants are indeed willing to sacrifice their satisfaction to save power. (3) In unstructured color transformation, color schemes generated by mono-color scheme usually have better user satisfaction than the ones generated by multi-color scheme.

The proposed methods can be employed at different stages of the lifetime of a mobile GUI. Faster and more efficient methods can be used at run-time by either the operating system or the application. Slower but more flexible methods can be used by GUI designers to identify designs that are both energy-efficient and aesthetically attractive.

ACKNOWLEDGEMENTS

The work was supported in part by NSF Award IIS/HCC 0713249 and by the Texas Instruments Leadership University program.

REFERENCES

[1] L. Zhong and N. K. Jha, "Graphical user interface energy characterization for handheld computers," in *Proc. Int. Conf. Compilers, Architecture, and Synthesis for Embedded Systems (CASES),* San Jose, California, USA, 2003.

[2] S. R. Forrest, "The road to high efficiency organic light emitting devices," *Organic Electronics,* vol. 4, pp. 45-48, 2003.

[3] S. Iyer, L. Luo, R. Mayo, and P. Ranganathan, "Energy-Adaptive Display System Designs for Future Mobile Environments," in *Proc. ACM/USENIX Int. Conf. Mobile Systems, Applications, and Services (MobiSys),* San Francisco, California, 2003.

[4] M. T. Raghunath and C. Narayanaswami, "User Interfaces for Applications on a Wrist Watch," *Personal Ubiquitous Computing,* vol. 6, pp. 17-30, 2002.

[5] P. Ranganathan, E. Geelhoed, M. Manahan, and K. Nicholas, "Energy-Aware User Interfaces and Energy-Adaptive Displays," *IEEE COMPUTER,* pp. 31-38, 2006.

[6] H. Shim, N. Chang, and M. Pedram, "A Backlight Power Management Framework for Battery-Operated Multimedia Systems," *IEEE Design & Test,* vol. 21, pp. 388-396, 2004.

[7] A. K. Bhowmik and R. J. Brennan, "System-Level Display Power Reduction Technologies for Portable Computing and Communications Devices," in *Proc. IEEE Int. Conf. Portable Information Devices*, Orlando, Florida, USA, 2007.

[8] W.-C. Cheng, C.-F. Hsu, and C.-F. Chao, "Temporal vision-guided energy minimization for portable displays," in *Proc. Int. Sym. Low Power Electronics and Design (ISLPED),* Tegernsee, Bavaria, Germany, 2006.

[9] T. Harter, S. Vroegindeweij, E. Geelhoed, M. Manahan, and P. Ranganathan, "Energy-aware user interfaces: an evaluation of user acceptance," in *Proc. ACM Conf Human Factors in Computing Systems (CHI),* Vienna, Austria, 2004.

[10] N. Kamijoh, T. Inoue, C. M. Olsen, M. T. Raghunath, and C. Narayanaswami, "Energy trade-offs in the IBM wristwatch computer," in *Proc. IEEE Int. Sym. Wearable Computers*, Zurich, Switzerland, 2001.

[11] 4D Systems, www.4dsystems.com.au/.

[12] M. Dong, Y.-S. K. Choi, and L. Zhong, "Power Modeling of Graphical User Interfaces on OLED Displays," in *Design Automation Conference (DAC),* San Francisco, CA, USA, 2009.

[13] Windows Mobile Themes, www.wm6themes.com/.

[14] B. Fraser, C. Murphy, and F. Bunting, *Real world color management*: Pearson Education, 2002.