LPBP: Low-Power Basis Profile of the Java 2 Micro Edition

Inseok Choi, Hyung Soo Kim, Heonshik Shin and Naehyuck Chang*
School of Computer Science & Engineering, Seoul National University, Korea
naehyuck@snu.ac.kr

ABSTRACT

The Java platform provides a fully fledged programming environment for graphics applications using the Abstract Window Toolkit (AWT). We present a power-aware basis profile of the Java 2 Micro Edition (J2ME) for embedded applications. The Low-Power Basis Profile (LPBP) is responsive to backlight luminance scaling in such a way that dynamic adjustment of the backlight luminance is accompanied by adaptive image compensation. The proposed scheme performs aggressive backlight dimming while maintaining the readability of screen contents based on image compensation techniques such as brightness compensation, image enhancement and context processing. Experiments show that on average the LPBP can easily achieve approximately 30% backlight system power reduction.

Categories and Subject Descriptors: B.4.2 [Input/Output and Data Communications]:Input/Output Devices–*Image display*; C.3 [Computer System Organization]:Special-purpose and application-based systems

General Terms: Design, Experimentation

Keywords: Low-power design, display, LCD, JAVA

1. INTRODUCTION

The Java runtime environment or its equivalent is now in common use, despite its inefficiency in execution, because it improves productivity and reduces the cost of software development and maintenance across most modern operating systems. Thus embedded software developers can design application programs with a Graphical User Interface (GUI) mat supports many different platforms. GUI implementation used to be difficult because graphics hardware and graphics toolkits were different for each platform. The Java AWT's platform-independent environment for graphics provides an attractive alternative and is now widely used in embedded platforms [1].

The Java 2 Micro Edition (J2ME) is also a suitable choice for embedded applications; J2ME is reasonably lightweight but preserves most of the advantages of the standard Java 2 Platform (J2SE) [2]. In this paper, we present a Low-Power Basis Profile (LPBP) of the Java 2 Micro Edition (J2ME) for embedded applications. In doing so, we implement the LPBP which extends the existing PBP in the J2ME architecture to enhance the visibility of an LCD with a dimmed backlight.

Hand-held embedded systems now require quality display systems such as color TFT LCDs. The display system components dominate the system-level power consumption while interactive graphical applications are running. Among these display system components, the backlight lamp is the greediest power consumer. Most low-power techniques adopt dynamic power management for the CPU, but there is no explicit idle time for the backlight as long as useful information is on the screen. Brute-force dimming of the backlight may used as a last resort, but is not desirable since it sacrifices display quality significantly. On the other hand, the backlight dimming accompanied by appropriate image compensation achieves significant power saving while maintaining screen readability [3]. This power saving is achieved by appropriate context processing, which helps compensate images using contextual information about the objects being displayed.

Building a low-power version of AWT starts with the implementation of lightweight AWT components using the building blocks provided by the current Personal Basis Profile (PBP). The Personal Profile (PP) has native heavyweight AWT components based on Xlib and thus is not applicable to hand-held systems, although the final goal of the Low Power Basis Profile or LPBP is to embed various low-power features including but not limited to the AWT.

2. JAVA 2 MICRO EDITION

Sun Microsystems announced the J2ME specifically for embedded applications: the J2ME is a downsized version of the J2SE. The J2ME defines the necessary *Configurations*, *Profiles* and *Packages* to build a complete Java runtime environment [2]. Fig. 1 shows the hierarchical structure of the J2ME. The Configurations are composed of a Virtual Machine (VM) and a minimal set of class libraries. The Profile complements the Configurations by adding additional classes, usually called the class libraries of the J2SE that provide an application program interface (API) for different types of target devices. The optional Packages allow further extension of the Profiles.

The J2ME focuses on home appliances and embedded devices, such as set-top boxes, gateways, mobile phones and high-end PDAs (Personal Digital Assistants). The J2ME supports two major configurations: the Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC). The CLDC is applicable to cell phones and low-end PDAs with around 512KB of available memory. The CLDC supports the minimum building blocks of the Java runtime environment on the Kilobyte Virtual Machine (KVM); but the CDC supports devices with more processing power and memory, at least 2 MB. The target platforms are high-end PDAs equipped with a 32-bit RISC processor and a network connection. The CDC includes full-featured Java VM with a set of core libraries. We exclude the CLDC from our discussion because a standard graphics library, which is justified with quality display environment, requires substantial processing power.

3. J2ME LOW-POWER PROFILE

3.1 Power consumption of the J2ME runtime environment

The baseline platform is equipped with a 32-bit RISC processor running at around 200MHz clock speed, a 64MB off-chip main memory, 8KB 2-way set-associative I and D caches, and a 4 inch 640×480 high color TFT LCD display with a CCFL backlight. We selected the Toshiba LTM04C380K transmissive TFT LCD panel to evaluate the actual power consumption [4].

We simulated the system-wide power consumption based on a precise energy model and cycle-accurate energy estimation [5]. On average, applications running on the J2ME runtime environment

*Corresponding Author. This work is partly supported by the Brain Korea 21 project. The Institute of Computer Technology at Seoul National University has provided some research facilities for this study.

ISLPED'03, August 25–27, 2003, Seoul, Korea.

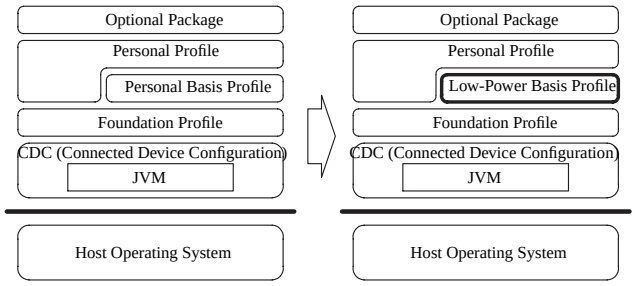

Figure 1: J2ME with Low-Power Basis Profile.

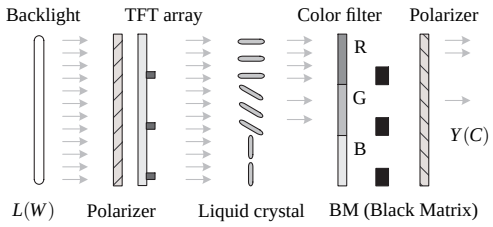

Figure 2: The structure of an LCD. The liquid crystals are differently rotated by AC voltages applied to the electrodes. The resulting rightness color values $A_j(C)$ vary linearly with the voltage.

exhibit a 99.9% I cache hit ratio, a 97% D cache hit ratio, represent 8% of task activity and draw 61440 pixels per second on the average. If we assume that the system embodies modern power reduction techniques for the CPU and memory systems that have come out of previous research, we found that the display components consume over 90% of the total power. This is accompanied for by the periodic refreshing of the display system and the power consumption of the backlight. Unlike the computational tasks, the display has no slack time that would allow for dynamic power management (DPM). Among the display components, the backlight consumes more than 1W power and occupies 40% of the total system power.

3.2 Low-power AWT

Fig. 1 illustrates a desirable location for the LPBP that allows the low-power features to be embedded within the J2ME. Since the Personal Basis Profile (PBP) of the J2ME is mainly composed of building blocks for the lightweight AWT components, designing the LPBP implies a low-power extension of the lightweight AWT.

The maximum energy gain is achieved when the backlight luminance dimming adjusts to every frame buffer update as the image histogram and the context information change (*i.e.*, dynamic adaptation of backlight luminance dimming). Choi and his colleagues [3] have also introduced a concept of dynamic adaptation in the low-power design contest; known as Dynamic Backlight Luminance Scaling (DLS). In summary, DLS reduces the backlight luminance followed by proper image compensation. Since its operation is dynamic, it performs image compensation and re-calculates the backlight luminance online.

DLS can be implemented at various levels of a design. As the implementation layer lies closer to the application, more aggressive backlight dimming is achievable by utilizing the context of the image. At the same time, the implementation of each application may be *adhoc* and proprietary. On the other hand, if we implement the layer close to the low-level hardware or the device driver, only a small amount of dimming is achievable due to lack of context information however, in this case, the DLS can be transparent to each application program. For the J2ME, the PBP-level design for low-power AWT provides application independence but also provides context information. The low-power AWT makes it possible to utilize context information based on the location and size of background and foreground data in the structure that recursively ascends to their parent containers. We embed image compensation algorithms such as brightness compensation, image enhancement and context processing in the LPBP.

4. IMAGE PROCESSING ALGORITHMS

4.1 Liquid crystal display and backlight

Our image compensation technique is based on the principle of backlight luminance dimming. As shown in Fig. 2, a white backlight generates the light source, and the liquid crystal associated with the color filter selectively filters the light [6]. The basic approach of the backlight luminance dimming is to reduce the luminance of the light source and make the liquid crystal screen block less of the light. This enhances the image luminance, so that human eyes recognize the same intensity (*i.e.* brightness) [3].

An LCD is an additive color reproduction system. A color C on an LCD can be matched by three primary colors R, G and B. The qualitative match of a color C is denoted by

$$C \equiv A_1 R + A_2 G + A_3 B \tag{1}$$

where A_1, A_2 and A_3 are the matching values of the color C. The luminance of the backlight is calculated by

$$L(v) = \sum_{j=1}^{3} \int_{\lambda} W_j P_j(v,\lambda) V(\lambda)\, d\lambda, \tag{2}$$

where W_j is the color matching value for the backlight, typically white in TFT LCDs, v is the supply voltage of the backlight lamp, $V(\lambda)$ is the relative luminous efficiency, and $P_j(v,\lambda)$ represents the spectral energy distribution of the three primary colors. The luminance of a pixel, whose color value is C, is denoted by

$$Y(C,v) = \gamma\, L(W,v) \sum_{j=1}^{3} \rho_j A_j(C), \tag{3}$$

where γ is the product of the aperture ratio and the transmittance of the two polarizers (Fig. 2) and ρ_j is the transmittance values of the three color filters. The aperture ratio is determined by the area of the black matrix (BM).

Note that, $A_j(C)$ denotes the normalized color values. Typical transmittance values of the color filter and the polarizer are $1/3$ and $1/2$ respectively. Modern LCDs have an aperture ratio of around $2/3$ [6].

4.2 Brightness compensation

Brightness, I, is the intensity, as perceived by the eye. The human perception system is beyond the scope of this paper, and we assume without loss of generality that the brightness is linearly proportional to the luminance of the LCD panel (*i.e.* $I = \alpha Y$ where α is a positive real number). This is because an LCD is a matched, additive color reproduction system and we may assume that the normalized color value, $A_j(C)$, is linearly proportional to the cone signal of the human eye.

Let $\mathbf{C}_i$ be the ith image with a vertical and horizontal resolution of V and H, respectively.

$$\mathbf{C}_i = \begin{pmatrix} C^i_{1,1} & \cdots & C^i_{1,H} \\ \vdots & \ddots & \vdots \\ C^i_{V,1} & \cdots & C^i_{V,H} \end{pmatrix}. \tag{4}$$

Let $L(v)$ and $L(v')$ be the luminance values of the original backlight and the dimmed backlight, respectively. Our goal is to maintain the luminance of $Y(\mathbf{C}_i, v)$ with respect to a change in backlight luminance from $L(v)$ to $L(v\prime)$, such that

$$Y(\mathbf{C}_i, v) \approx Y(\mathbf{C'}_i, v'). \tag{5}$$

From Eq. 3, we obtain

$$A_j(\mathbf{C}'_i) \approx \frac{A_j(\mathbf{C}_i)L(v)}{L(v')}. \quad (6)$$

If $A_j(C'_{k,l}) \leq 1$ for all $C'_{k,l}$, where $1 \leq k \leq H$ and $1 \leq l \leq V$, then $Y(\mathbf{C}_i, v) = Y(\mathbf{C}'_i, v')$ (*i.e.*, perfect brightness compensation). However, in practice, if $\exists C'_{k,l}$, such that $A_j(C'_{k,l}) > 1$, perfect compensation cannot be achieved. This happens as v' is significantly reduced, or large number of pixels occur in bright areas. The overflow, $A_j(C'_{k,l}) - 1$, is truncated and this causes distortion. We use a histogram function, $H_j(\mathbf{M}_i)$, to quantify the distortion ratio, D_i, which is denoted by

$$D_i = \frac{\sum_{j=T_H}^{2^n-1} H_j(\mathbf{M}_i)}{\sum_{j=0}^{2^n-1} H_j(\mathbf{M}_i)}, \quad (7)$$

where n is the color depth. The matrix $\mathbf{M}_i$ is a $V \times H$ matrix such that

$$M^i_{k,l} = 2^n \max(A_1(C^i_{k,l}), A_2(C^i_{k,l}), A_3(C^i_{k,l})) \quad (8)$$

in an RGB-color space. We can derive $\mathbf{M}_i$ from YUV and HSV color spaces if we accept minor errors. In a YUV-color space

$$M^i_{k,l} = 2^n Y(C^i_{k,l}). \quad (9)$$

In an HSV-color space, $M^i_{k,l} = 2^n V(C^i_{k,l})$.

Brightness compensation is implemented by a color transformation function such that

$$c' = T_{bc}(c) = \min(2^n - 1, \frac{2^n - 1}{T_H} c). \quad (10)$$

The threshold, T_H is determined by the given value of D_i and by Eq. 7. In practice, only pixels in the brightest area are distorted after compensation while others keep their original colors and brightness.

4.3 Image enhancement

Image enhancement does not attempt to preserve the original color values exactly under more aggressive backlight dimming. But it does make the image more recognizable with a dimmer backlight. We introduce histogram stretching and histogram equalization for that purpose.

Histogram stretching is a well-known image enhancement technique in the spatial domain [7, 8, 9, 10]. Histogram stretching is a mixture of contrast stretching and thresholding. The distortion ratios, D^L_i and D^H_i, in histogram stretching are defined as shown below:

$$D^H_i = \frac{\sum_{j=T_H}^{2^n-1} H_j(\mathbf{M}_i)}{\sum_{j=0}^{2^n-1} H_j(\mathbf{M}_i)}, \qquad D^L_i = \frac{\sum_{j=0}^{T_L} H_j(\mathbf{M}_i)}{\sum_{j=0}^{2^n-1} H_j(\mathbf{M}_i)}, \quad (11)$$

where T_L and T_H are the low and the high thresholds, respectively, determined by the given distortion ratio, D^L_i and D^H_i. The transformation function is given by

$$c' = T_{hs}(c) = \min(2^n - 1, \frac{2^n - 1}{T_H - T_L} \max(0, (c - T_L))), \quad (12)$$

The use of histogram equalization has a general tendency to spread the histogram of the original image, so that the levels of a histogram-equalized image span a fuller range in terms of the gray scale values [8]. Histogram equalization is defined by a transformation function as follows:

$$c' = T_{he}(c) = 2^n \frac{\sum_{j=0}^{c} H_j(\mathbf{M}_i)}{\sum_{k=0}^{2^n-1} H_k(\mathbf{M}_i)}. \quad (13)$$

When we separately apply Eq. 13 to each primary color domain, erroneous colors may be generated [8]. Thus, we calculate Eq. 13 in the form of a look-up table in the *Value* domain of the HSV color model.

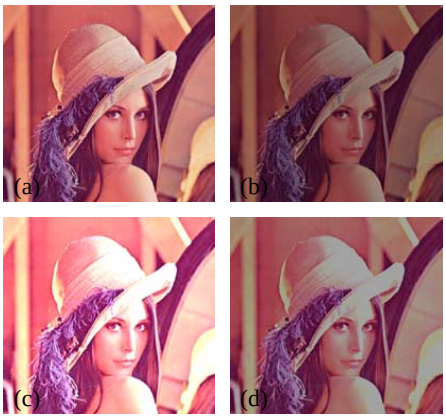

Figure 3: Brightness compensation. (a) original image; (b) simulation of the original image under 64% backlight luminance; (c) brightness compensation with $D_i = 0.7$; and (d) simulation of brightness compensation under 64% backlight luminance.

4.4 Context processing

So far, we have concerned ourselves with context-free image compensation and enhancement techniques to maintain image quality under a dimmed backlight. Context processing aims for extreme backlight dimming by re-mapping the original colors.

Since the context processing algorithm is embedded in the Profile of the J2ME, the context information is delivered without extra overhead. We need to ensure that the foreground and background colors remain complementary color, mapping regardless of the original colors, when the backlight dimming is extreme. This offers the possibility of more aggressive backlight dimming even though the improvement obtainable from brightness compensation and/or image enhancement have reached their limit.

Context processing is effective in two different cases: *a)* the developers intentionally assign similar colors to the foreground and the background objects for a specific look and feel or *b)* the foreground and the background colors are similar after the brightness compensation and/or image enhancement. The transformation function is given by

$$c' = T_{cp}(c) = 2^n - b(c) - 1 \quad (14)$$

where $b(c)$ is a background color of an object of color c.

4.5 Backlight Luminance Control

Most color TFT LCD systems allow control of the backlight luminance. This is used to the extended battery life, depending on the user's preference. White LED backlight systems are responsive, and open-loop control with no feedback to enhance the response time is adequate to support DLS. So we can directly implement DLS-aware AWT without any hardware modification of the LED backlight system.

On the other hand, currently cold cathode fluorescent lamps (CCFL) are much more popular due to cost and ease of manufacture, but they are too slow to support DLS. But simple feedback control such as PI (proportional and integral) control will ad-

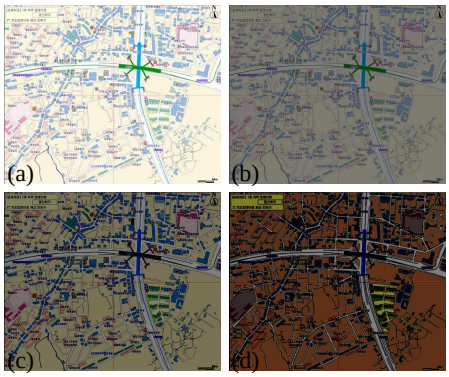

Figure 4: Image enhancement. (a) Original image; (b) simulation of the original image under 50% backlight luminance; (c) simulation of histogram stretching with 10% threshold under 50% backlight luminance; and (d) simulation of histogram equalization under 50% backlight luminance.

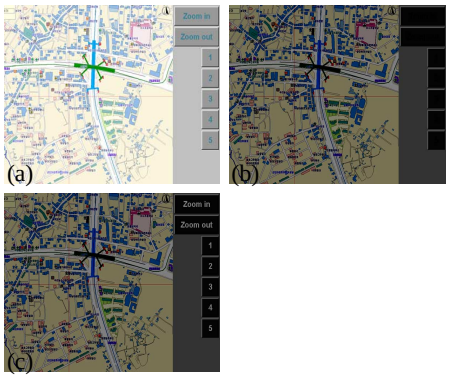

Figure 5: Context processing. (a) Original image; (b) simulation of histogram stretching with 10% threshold; and (c) simulation of context processing after histogram stretching with 50% backlight luminance.

equately enhance the response time. In this work, we use a CCFL backlight system and add a small microcontroller with an 8-bit register and a photo sensor. Using this feedback controller, we have achieved a 20ms response time. The luminance can be scaled from 0 to 255.

5. EXPERIMENTAL RESULTS

The low-power AWT is useful to extend the battery life in two situations. First, the user may want longer battery life without appreciable degradation of the display quality. Second, the user may want the maximum battery life, and will accept noticeable degradation of the display quality as long as the screen contents are still recognizable. Brightness compensation is applicable to the first situation. Image enhancement and/or context processing are suitable for the second situation. Since there is no standard metric to compare display quality with distortion ratio, we perform a MOS (mean opinion score) test to determine the minimum luminance of the backlight for each application.

We have implemented the hardware system for the J2ME LPBP with a 6.4-inch color TFT LCD. However, for better presentation, figures in this paper are simulated by Eq. 3 instead of photographing the LCD panel.

5.1 Brightness compensation

The relative backlight luminance is determined by a given distortion ratio, D_i, following Eqs. 6 and 7. Fig. 3 shows the results of backlight dimming with image compensation and the resultant power reduction. The power reductions of the backlight and overall system are 18.8% and 8.3% respectively.

5.2 Image enhancement

Fig. 4 shows examples of image enhancement and resulting power reduction. The experiments demonstrate that image enhancement generally allows within 50% of backlight luminance dimming while retaining acceptable recognizability. But the amount of backlight dimming is solely dependent on user preference, which is quite subjective. The luminance of the backlight is 50% of the original and thus there is a 50% saving in backlight power.

5.3 Context processing

Fig. 5 demonstrates that, if a frame contains either pictures or AWT components, we can use either brightness compensation or image enhancement. But, using brightness compensation or image enhancement on a frame that consists of *both* pictures and components may result in undesirable results, as shown in Fig. 5-(b). Image enhancement results in poor perception, as the background and the foreground colors become similar to each other after processing. The process of determining the threshold for histogram stretching is dominated by the picture; so the background and the foreground colors of the AWT components are not correctly changed.

Context processing complements this undesirable results by changing the foreground color to complement the background. Fig. 5-(c) demonstrates the achievement of 50% backlight-power reduction using context processing.

6. CONCLUSIONS

This paper introduces lightweight AWT components of the Low-Power Basis Profile for the Java 2 Micro Edition as an extension of the Personal Basis Profile. The Low-Power Basis Profile includes lightweight AWT components for use in backlight dimming. The power-aware features are derived from system-wide power analysis of a hand-held embedded system with a color TFT LCD running typical GUI applications in the J2ME runtime environment, in which more than 40% system power goes to the backlight. Backlight dimming must involve image compensation to maintain the display quality.

We have devised image compensation algorithms for backlight dimming which consider the context of the AWT components. Brightness compensation is suitable for high fidelity displays with regular backlight dimming, and demonstrate 18% backlight power reduction. Experimental results show a backlight-power reduction is 31% and 36%, for image enhancement with aggressive backlight dimming and for context processing with extreme backlight dimming, respectively.

7. REFERENCES

[1] D. M. Geary, *Graphic Java 1.2, Mastering JFC*. Sun Microsystems Press, San Francisco, CA, USA, 1999.
[2] *Java 2 Platform, Micro Edition*, 2002. [Online]. Available: http://java.sun.com/j2me.
[3] I. Choi, H. Shim, and N. Chang, "Low-power color TFT LCD display for hand-held embedded systems," in *Proceedings of International Symposium on Low Power Electronics and Design*, August 2002, pp. 112–117.
[4] *LTM04C380K, Product Information*, 2000. [Online]. Available: http://www.tridentdisplays.co.uk/cgi-gen/nph-request.pl?LTM04C380Kv10.pdf
[5] N. Chang, K.-H. Kim, and H. G. Lee, "Cycle-accurate energy measurement and characterization with a case study of the ARM7TDMI," *IEEE Transactions on VLSI Systems*, vol. 10, pp. 146 – 154, Apr. 2002.
[6] T. Tsukada, *TFT & LCD*. Books Hill, March 2000.
[7] W. K. Pratt., *Digital Image Processing*, 3rd ed. New York, NY, USA: John Wiley & Sons, 2001.
[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Reading, MA, USA: Addison-Wesley, 1992.
[9] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice Hall, 1989.
[10] W. K. Pratt, *Digital Image Processing*. A Wiley-Interscience Publication, 2001.