Dynamic Backlight Adaptation for Low-Power Handheld Devices

Sudeep Pasricha, Manev Luthra, Shivajit Mohapatra, Nikil Dutt, and Nalini Venkatasubramanian
University of California, Irvine

Editors' note:
Backlight power minimization can effectively extend battery life for mobile handheld devices. This article proposes an adaptive middleware-based approach to optimize backlight power consumption when playing streaming video. The technique simultaneously minimizes the negative impact on perceived video quality.

—Radu Marculescu, Carnegie Mellon University; and Petru Eles, Linköping University

MODERN HANDHELD MOBILE DEVICES of modest size and weight increasingly provide users with streaming multimedia content. These devices have limited computing, storage, and battery resources, yet the multimedia applications they run tend to be extremely resource hungry. Typically, the display, the CPU, and the network interface are the three primary sources of power consumption in such low-power devices. Our work focuses on achieving energy savings from the devices' backlight display—without significantly compromising the streamed video quality. We performed this work in the larger framework of the Forge project,[1] which targets power and quality-of-service optimizations across multiple abstraction layers, including the application, middleware, operating system, and hardware.

In recent years, researchers have proposed several schemes to optimize power consumption by low-power devices in mobile environments. Techniques such as compiler-, operating-system-, and middleware-based adaptation;[1-3] dynamic power management of network interfaces[4] and disks;[5] and dynamic voltage scaling[6,7] attempt to reduce power consumption at various computational levels. However, there has been relatively little interest in efforts to reduce backlight power consumption. Choi, Shim, and Chang[8] proposed compensating the brightness of still images while simultaneously reducing the backlight level. Besides being limited to still images, their proposed contrast compensation doesn't preserve the image's original color, which limits the scheme's practical application.

Our work explores more aggressive approaches to brightness compensation and device backlight control for streaming video. Furthermore, we shift the adaptation away from the handheld device and perform it at a network proxy server, obviating the need to modify the decoder on the device. We find that aggressive brightness compensation without significant impact on visual quality is more practical for streaming video than for still images. Aggressive compensation introduces noticeable artifacts in still images, but these are less discernable in video because several frames appear on the screen every second. We also propose an effective brightness compensation algorithm and integrate it with our middleware-based adaptation schemes to achieve low-power backlight operation while streaming video content to mobile handheld devices.

System architecture

We implemented the system model shown in Figure 1. In addition to a video server and low-power wireless devices that can display streaming MPEG video content, the system entities include a proxy server with access to

- a database of profiled luminosity values for every video stream at the server;
- handheld-device-specific parameters (average lumi-

0740-7475/04/$20.00 © 2004 IEEE

Copublished by the IEEE CS and the IEEE CASS

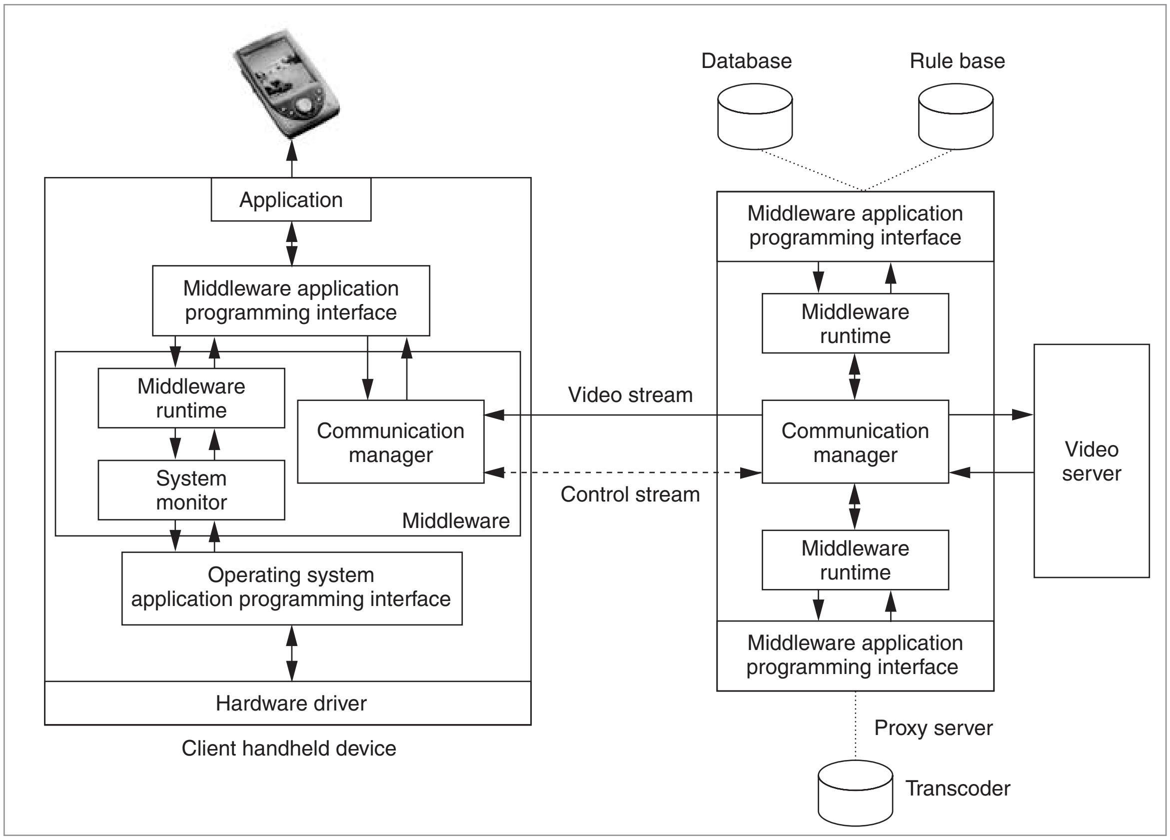

Figure 1. System architecture.

nosity at each level, number and average power consumption of backlight levels, and so on);

- a rule base to determine compensation values; and
- a video transcoder.

Moreover, all communication between the handheld devices and the video server passes through the proxy server, which can change the video stream in real time.

Each client has a middleware layer that routes the information flowing to and from the video decoder application, and an application layer where the video decoder application decodes the video stream. The communication manager module sends video requests from the applications to the proxy server and also sends information about device-specific features (such as the number of backlight levels, the current backlight level, and the type and make of the handheld device) gathered by the system monitor module. Using the appropriate API calls to the underlying operating system, the system monitor can also direct the handheld device to change backlight levels. The communication manager also receives video streams from the proxy server and directs them to the application layer for display on the handheld device's screen.

The proxy server's middleware dynamically adapts the streaming video content (brightness and contrast compensation) and communicates control information (operating backlight levels) to the client middleware through the communication manager module, which interfaces with the low-bandwidth control stream. The client uses the control stream to send the video stream request to the proxy, along with the current backlight level and other handheld-specific information, and the proxy uses the control stream to direct the client handheld device to set a new backlight level while playing streaming video. The proxy maintains a database of information about the videos available at the server, along with information specific to different handheld device types. Additionally, the proxy uses a static rule base specifying conditions that determine values for backlight and video compensation. Extensive profiling and subjective assessment of videos on different handheld devices populate the database and certain parameters of the rule base.[1]

Backlight compensation

Averaging for several different MPEG-1 video streams, we measured power dissipated at various backlight levels for the Compaq iPAQ handheld device running Windows CE. For the iPAQ's five backlight levels (super bright, high bright, medium bright, low bright, and power save), the average power dissipated was 2.80 W, 2.51 W, 2.32 W, 2.16 W, and 1.72 W, respectively. The results indicate that operating the device at a lower backlight intensity level produces significant energy savings.

The inherent problem with attempting to reduce power is that the backlight directly affects the display quality and the user experience. For example, even a slight reduction in backlight intensity to save power during multimedia playback on the handheld device can degrade the human perception of quality. Indeed, simply reducing the backlight is not a viable solution. Instead, we explored the use of a video compensation algorithm that induces power savings without noticeably affecting video quality.

Our prior work focused on several aspects of video quality in handheld devices.[1] We determined that the environment and the type of handheld device in use significantly influence user perception of video playback quality. Consequently, objective assessment[9] of video quality is extremely difficult, leaving subjective assessment[10] as the primary method. In our profiling and surveys, we tried to follow a relevant International Telecommunication Union recommendation,[10] and we chose a diverse collection of video streams (movie clips, animations, sports, documentaries, and so on) to assess our schemes' suitability and effectiveness.

To validate our assessments, we selected 30 individuals to be part of an extensive survey to subjectively assess video quality when viewing streaming video on a handheld device. We first showed the subjects a full-screen version of an original unaltered video stream. Next, we showed them the compensated stream and asked them to record their perceptions of differences in the video quality. We then tried several different combinations of parameters from our compensation algorithm to determine the values that would save the most power without perceptibly degrading video quality. We repeated this phase for several different video streams and recorded feedback from the subjects.

After extensively analyzing the feedback, we arrived at a set of values for the rule base, luminance thresholds, and parameters in our compensation algorithm that give us substantial power savings with acceptable degradation in video quality. We do not intend to sacrifice video quality to reduce handheld device power consumption.

Dual-compensation algorithm

Here we characterize the problem and propose our compensation algorithm. Let n be the number of backlight levels supported, and $P(n)$ be the power at each level. Then $P_{\text{save}} = P(n) - P(n-k)$ denotes the power savings when dimming the backlight from level n to $n-k$. An image's perceived intensity is denoted by

$$I = \rho L Y_{\text{fr}}$$

where ρ is the transmittance of the LCD panel, L is the backlight luminance at level n, and Y_{fr} is the frame's average luminance value.[8] We can obtain a pixel's luminance value (Y_{pix}) from its RGB values after applying standard conversion functions to convert it from RGB to the $\text{YC}_\text{r}\text{C}_\text{b}$ coordinate space.[11] Let L' give the luminance at level $n-k$. According to Choi, Shim, and Chang,[8] if we decrease the backlight level from n to $n-k$, then to preserve the perceived intensity, we derive the new luminance value Y_{pix}' for each pixel in the frame by

$$\xi(Y_{\text{pix}}') = \min(1, \xi(Y_{\text{pix}}) + \Delta L)$$

where $\xi(Y_{\text{pix}}')$ is the normalized value of Y_{pix}', and

$$\Delta L = (1 - L'/L)$$

However, we cannot adequately compensate pixels already having a high luminosity value; therefore, we lose contrast because of saturation and observe degradation in video quality. This limits the amount of compensation applicable to an image without degrading its quality intolerably. However, as we will show, there are ways to address and compensate for the loss in contrast, allowing even further savings in power resulting from a larger reduction in backlight intensity.

Working with groups of scenes

The concept of a group of scenes (GOS) lets us define the granularity at which we perform backlight compensation. We define a GOS as a group of contiguous frames in a video stream such that the variance of the average luminosity values of each frame belonging to the group is less than a threshold value α. We calculate a frame's average luminosity value Y_{fr} as

$$Y_{\text{fr}} = \sum_{i=0}^{(w-1)(h-1)} \frac{Y_{w,h}}{wh}$$

where w and h are the image's width and height in number of pixels, and $Y_{w,h}$ is the pixel's luminosity at (w, h). We use the GOS concept to split a video stream into several groups of frames that form the basic entities on which our algorithm performs compensation. Generally, in video streams, many frames with similar average luminosity values are clustered together, providing ample scope to optimize for low power by uniformly compensating entire GOS entities and reducing the handheld's backlight level. A GOS must have a minimum number of frames ß to be eligible for compensation. We introduce this parameter because there must be a minimum duration between changes in backlight levels.

We introduce function Ω, which provides the backlight level compensation factor:

$$\Omega(k_i, \Gamma, Y_{\text{gos}}) = k'$$

The input parameters to Ω are the current backlight level (k_i), the type of handheld device (Γ), and the average luminosity of the GOS under consideration (Y_{gos}) for which the number of frames is greater than ß, and the function returns the optimal backlight level to be set for the GOS (k'). The average GOS luminosity value appears as

$$Y_{\text{gos}} = \sum_{i=0}^{n-1} \frac{Y_{\text{avg}}}{n}$$

where n is the number of frames in the GOS. Using extensively profiled video information pertaining to a particular type of handheld device and stored in the proxy's database, this function selects and returns a suitable value for backlight level on the basis of the Y_{gos} and k_i values.

Next we introduce function σ, which provides the video luminosity compensation factor:

$$\sigma(k_i, \Gamma, \Omega(k_i, \Gamma, Y_{\text{gos}})) = c'$$

The input parameters for σ are the current backlight level (k_i), the type of handheld device (Γ), and the value for the next backlight level (returned by function Ω). Function σ returns the brightness compensation value for the GOS under consideration. The returned value c' is the difference in luminosity for the two backlight levels for the particular handheld device serving as a client. The proxy's database stores default values for several handheld devices and provides the average luminosity values for different backlight levels that the devices support. If k_i is the current backlight level on a handheld of type Γ, and Y_{gos} is the average luminosity of an eligible GOS streaming to the proxy from the video server, then a control message sent to the client asks it to set its backlight level to

$$\Omega(k_i, \Gamma, Y_{\text{gos}}) = n_i$$

while the GOS is compensated with a brightness of

$$\sigma(k_i, \Gamma, \Omega(k_i, \Gamma, Y_{\text{gos}})) = c'$$

before the proxy sends it to the client. This results in power savings of $P_{\text{save}} = P(k_i) - P(n_i)$ during the time the GOS plays back on the client.

Balancing contrast and luminosity

Increasing a frame's luminosity can cause a loss in contrast between different regions, making it difficult to identify edges of objects in the frame and degrading picture quality. To overcome this degradation, we propose an additional compensation step in which the luminosity-compensated frame passes through a high-pass filter that performs a spatial convolution on the frame's luminosity values. This convolution step sharpens edges and makes objects in the frame more recognizable. A convolution kernel takes the form

$$\mathbf{C}_{\boldsymbol{k}} = \begin{matrix} c_1 & c_2 & c_3 \\ c_4 & c_{\text{pix}} & c_5 \\ c_6 & c_7 & c_8 \end{matrix}$$

where $c_1, c_2, \ldots, c_8$ are values carefully selected to increase the amplitude of the frame's high-frequency content. The convolution kernel is not limited to a 3 × 3 matrix and can be larger (for example, 5 × 5). Now let the luminosity value of the pixel under consideration be L_{pix} and that of its eight neighboring pixels be

$$\boldsymbol{L}_{3\times 3} = \begin{matrix} L_1 & L_2 & L_3 \\ L_4 & L_{\text{pix}} & L_5 \\ L_6 & L_7 & L_8 \end{matrix}$$

Then we show the modified pixel value after convolution as

$$L_{\text{pix}}' = \frac{1}{\sum_{k=1}^{8} c_k}\left(L_{\text{pix}} c_{\text{pix}} + \sum_{k=1}^{8} (L_k c_k)\right)$$

$$\text{if } c_{\text{pix}} + \sum_{k=1}^{8} c_k > 0$$

and

$$L_{\text{pix}}{}' = \left(L_{\text{pix}} c_{\text{pix}} + \sum_{k=1}^{8} (L_k c_k) \right) + 128$$

$$\text{if } c_{\text{pix}} + \sum_{k=1}^{8} c_k = 0$$

An alternative to the high-pass filter is a median filter—that is, a nonlinear filter that not only sharpens a frame but is also noise tolerant. This is generally more difficult and time-consuming to implement than the high-pass filter we described. Our experiments have shown that this particular nonlinear filter produces images of the same quality as those produced by the high-pass filter for the video streams considered.

Proxy-based middleware adaptation schemes

There are three middleware adaptation policies that use the compensation algorithm. It's possible to implement the first two policies with limited control of the operating system interface. The third policy uses our proposed dual-compensation algorithm, which requires both proxy and operating system interfaces for optimal operation.

Simple backlight compensation

Using the simple backlight compensation (SBC) policy, we can save power by identifying GOS entities for which Y_{gos} is high (above a threshold level, τ) and reducing the backlight level at the client for these GOS entities. Note that the proxy simply calculates new backlight levels without compensating for the video stream's brightness. The proxy has a record of the GOS entities in the video stream, and just before it sends a GOS with $Y_{\text{gos}} > \tau$, it sends control information to the client to reduce its backlight level, so that the perceived difference in quality is minimal. The disadvantage of this scheme is that it degrades video quality in every case; it's attractive only because of its simplicity (it performs no video compensation).

Table 1. Parameter values for compensation algorithm used in experiments.

Parameter	**Value**
Minimum no. of frames, β, in GOS	60
Variance threshold, α, for Y_{gos}	40
Threshold level, τ, for SBC scheme	220
Convolution kernel, c_k, used in high-pass filter	$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$

Constant backlight with video luminosity compensation

A more interesting and practical approach is constant backlight with video luminosity compensation (CBVLC), whereby we set a constant backlight level at the start of the video stream and then dynamically compensate different GOS entities on the basis of their Y_{gos} values. Because the backlight value is fixed for the video's entire duration, it's essential to choose the level conservatively so that dramatic variations in luminous intensity of consecutive GOS entities in the video stream don't adversely affect video quality. This static approach, with no dynamic adaptation of backlight intensity levels, would save a constant (but limited) amount of power, depending on the initial backlight level chosen.

Dual-compensation approach

A hybrid dual-compensation approach (DCA) can overcome the limitations of the previous approaches. With this technique, we simultaneously compensate the video stream and the backlight levels for different GOS entities. The proxy dynamically compensates the GOS entities in the video stream and begins streaming the video to the client, simultaneously directing the client, through the control stream, to change its backlight level. The client middleware sets the appropriate backlight intensity levels for the video playback. This approach provides more flexibility for aggressive optimizations and results in far greater power savings.

Performance evaluation

To evaluate these approaches, we experimented with several different MPEG-1 video streams and a Compaq iPAQ 3600 series mobile handheld device, which comes with a color-reflective, thin-film-transistor LCD screen with a pixel pitch of 0.24 mm and a display resolution of 240 × 320 pixels.

Table 1 shows the parameters we used for our compensation algorithm. In addition, we show the rule base for determining $\Omega(k_x)$ and $\sigma(k_x)$ for the Compaq iPAQ's five backlight levels (k_0, k_1, k_2, k_3, and k_4) as follows:

Table 2. Characteristics of video streams used in the experiment.

Name	Resolution (pixels)	Frames per second	Duration (seconds)	Luminosity variation	Video type
bipolar.mpg	320 × 240	30	41	Little	Dark, 3D animation
iceegg.mpg	240 × 136	30	59	Moderate	Bright, 3D animation
intro.mpg	160 × 120	30	59	Very high	Flashy, TV show clip
simpsons.mpg	192 × 144	30	27	High	Colorful, 2D animation

$$\begin{aligned}
\Omega(k_0) &= k_0;\ \sigma(k_0) = 0; && \text{for all } Y_{\text{gos}}\\
\Omega(k_1) &= k_1;\ \sigma(k_1) = 0; && \text{for all } Y_{\text{gos}}\\
\Omega(k_2) &= k_1;\ \sigma(k_2) = 30; && \text{for } Y_{\text{gos}} < 140\\
\Omega(k_3) &= k_2;\ \sigma(k_3) = 30; && \text{for } 80 < Y_{\text{gos}} < 190\\
&= k_1;\ \sigma(k_3) = 55; && \text{for } 190 < Y_{\text{gos}} < 220\\
\Omega(k_4) &= k_3;\ \sigma(k_4) = 30; && \text{for } 190 < Y_{\text{gos}} < 220\\
&= k_2;\ \sigma(k_4) = 55; && \text{for } 60 < Y_{\text{gos}} < 190\\
&= k_1;\ \sigma(k_4) = 65; && \text{for } Y_{\text{gos}} < 60
\end{aligned}$$

We determined the rule base, the luminance threshold values, and the compensation algorithm parameters in Table 1 after extensive profiling and subjective assessment of several video streams,[1] because no analytical method exists for deriving them. We chose the values to achieve substantial power savings without significant degradation in video quality and stored these values in the proxy's database. Our compensation algorithm queries these parameters for different GOS entities to determine how much compensation to perform. Our approach is not limited to the Compaq iPAQ and is valid for other handhelds with fewer or more backlight levels (for example, the Palm Tungsten and HP Jornada handheld series). Each handheld type would have its own unique rule base, luminance threshold values, and compensation algorithm parameters, calculated through extensive profiling and assessment, just as we did for the iPAQ.

We analyzed the effectiveness of the three schemes for several video streams of different durations (number of frames), resolutions (frame width and height), and types (containing dynamically or statically changing scenes). To conserve space, we present the results for a limited set of streams in Table 2. In all our experiments, we assumed that the handheld's initial brightness level takes into account the brightness of the client's environment such that the user finds the video playback quality acceptable at that level. Our schemes are not applicable if the initial backlight setting is low or adjusted to a power-save setting; the rule base recognizes these low settings. We used a National Instruments peripheral component interconnect (PCI) data acquisition board to sample voltage drops across a resistor and the iPAQ, and we sampled the voltages at 200,000 samples per second to determine the iPAQ's variations in power consumption. We calculated the average instantaneous power using the equation

$$P = V_{\text{iPAQ}}(V_R / R)$$

where V_{iPAQ} is the average voltage drop across the iPAQ, V_R is the average voltage drop across a resistor connected in series with the iPAQ, and R is a 0.22-ohm resistor.

The first experiment assumes that the handheld device's initial backlight level is set to super bright. Figure 2a shows the results for this case. It's easy to see that the DCA scheme outperforms the other two schemes in all cases. The CBVLC scheme outperforms the simplistic SBC scheme in all cases except for intro.mpg. This video has a very high variation of average luminous intensity from one frame to the next, and very high intensity values for some frames. Consequently, very few GOS entities are eligible for compensation.

Because the CBVLC scheme sets the client backlight level just once (in the beginning), lowering the backlight level can significantly degrade the quality for many frames in this case, and compensation cannot rectify it. Therefore, the CBVLC scheme takes the conservative approach and doesn't ask the client to lower its backlight level from its preset value. Interestingly, in the bipolar.mpg case, the SBC scheme doesn't save any power. This is because, on average, the video is very dark (low luminance), and no GOS entity has a value of $Y_{\text{gos}} > \tau$. Consequently, in this scheme the backlight level is never reduced for this video.

Figures 2b and 2c show the results when we assume that the handheld device's initial backlight level is set to high bright and medium bright, respectively. As we lower the initial backlight levels, we expect the scope for reducing power consumption in the handheld to

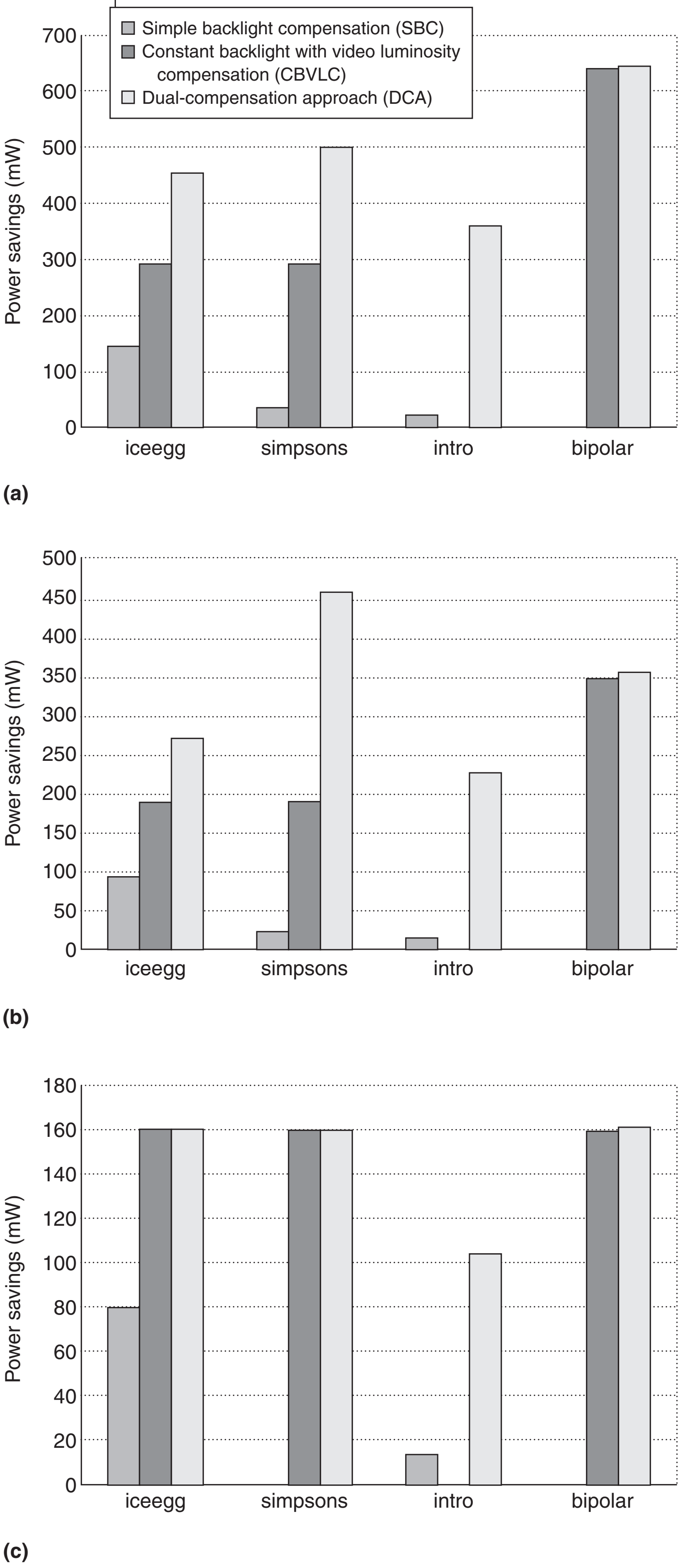

Figure 2. Power savings for initial reference points of super bright (a), high bright (b), and medium bright (c), by adaptation scheme.

decrease, which the results show by lower values on the *y*-axis. For the experiment with the initial level set to medium bright (Figure 2c), the CBVLC and DCA schemes perform more or less the same. The intro.mpg case is an exception (for the reasons given earlier).

The two schemes perform similarly because the DCA scheme can reduce the backlight by at most only one level. Further reduction makes it difficult to maintain quality, even with video compensation. Our rule base reflects this fact. The CBVLC scheme also lowers the backlight by one level for all cases except intro.mpg—hence, the performance similarity. These experiments show that the DCA scheme performs just as well as the other two schemes in a few cases but much better in the majority of cases. The DCA scheme's power savings ranges from 100 mW to 625 mW, depending on the type of video and the initial backlight setting. This corresponds to roughly a 9% to 60% reduction in power consumption by the mobile handheld client's backlight.

WE INTEND TO APPLY the proposed power-saving techniques to more handheld devices than just the Compaq iPAQ. Future research will focus on quantitatively analyzing video quality and developing analytical models that distinguish between acceptable and unacceptable video quality. This will let us automatically determine optimal compensation parameters, which until now we have determined through subjective assessment. Developing the analytical model, however, is challenging because video quality is inherently a subjective trait. For instance, an analytical model might look at peak signal-to-noise ratio for two frames and find them equivalent, but viewers might see the first frame as more acceptable because it has distortion somewhere in the background, while distortion in the second frame affects, say, a person's face and is therefore immediately apparent. ■

Acknowledgment

Funding from National Science Foundation NGS award ACI-0204028 supported this work.

■ References

1. R. Cornea et al., *Integrated Power Management for Video Streaming to Mobile Handheld Devices*, CECS tech. report 03-19, Univ. of California, Irvine, 2003.
2. S. Mohapatra and N. Venkatasubramanian, "PARM: Power-Aware Reconfigurable Middleware," *Proc. 23rd Int'l Conf. Distributed Computing Systems* (ICDCS 23),

IEEE CS Press, 2003, pp. 312-321.

3. B.D. Noble et al., “Agile Application-Aware Adaptation for Mobility,” *Proc. 16th ACM Symp. Operating Systems Principles* (SOSP 97), ACM Press, 1997, pp. 276-287.
4. S. Chandra, “Wireless Network Interface Energy Consumption Implications of Popular Streaming Formats,” *Proc. Multimedia Computing and Networking* (MMCN 02), ACM Press, 2002, pp. 185-201.
5. F. Douglis, P. Krishnan, and B. Marsh, “Thwarting the Power Hungry Disk,” *Proc. Winter Usenix Conf.*, Jan. 1994, Usenix Assoc., pp. 292-306.
6. C.J. Hughes et al., “Saving Energy with Architectural and Frequency Adaptations for Multimedia Applications,” *Proc. 34th Ann. Int’l Symp. Microarchitecture* (MICRO-34), ACM Press, 2001, pp. 250-261.
7. P. Pillai and K.G. Shin, “Real Time Dynamic Voltage Scaling for Low Power Embedded Operating Systems,” *Proc. 18th ACM Symp. Operating Systems Principles* (SOSP 01), ACM Press, 2001, pp. 89-102.
8. I. Choi, H. Shim, and N. Chang, “Low-Power Color TFT LCD Display for Hand-Held Embedded Systems,” *Proc. Int’l Symp. Low Power Electronics and Design* (ISLPED 02), ACM Press, 2002, pp. 112-117.
9. J. Lubin, “A Human Vision System Model for Objective Picture Quality Measurements,” *Proc. Conf. Int’l Broadcasters’ Convention*, Int’l Broadcasters’ Convention, 1997, pp. 498-503.
10. *Recommendation ITU-R BT.500.7, Methodology for the Subjective Assessment of the Quality of Television Pictures*, Int’l Telecommunication Union, 1995.
11. R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communications and Applications*, Prentice Hall, 1995.

Sudeep Pasricha is a PhD candidate at the University of California, Irvine. His research interests include design space exploration of SoC communication architectures, system modeling methodologies, and low-power distributed embedded-system design. Pasricha has a BE in electronics and communications engineering from Delhi Institute of Technology, India, and an MS in computer science from the University of California, Irvine. He is a student member of the IEEE.

Manev Luthra is a PhD candidate at the University of California, Irvine. His research interests include embedded-systems design, low-power-systems design, and distributed systems. Luthra has a BTech in electrical engineering from the Indian Institute of Technology, Delhi. He is a student member of the IEEE.

Shivajit Mohapatra is a PhD candidate at the University of California, Irvine. His research interests include power-aware middleware frameworks, low-power systems, and distributed systems. Mohapatra has an MS in computer systems and networking from the University of California, Irvine.

Nikil Dutt is a professor of computer science and electrical engineering affiliated with the Center for Embedded Computer Systems at the University of California, Irvine. His research interests include embedded-systems design automation, computer architecture, compiler optimization, low-power design, and system specification techniques. Dutt has a PhD in computer science from the University of Illinois at Urbana-Champaign. He is a senior member of the IEEE.

Nalini Venkatasubramanian is an associate professor in the Department of Information and Computer Science at the University of California, Irvine. Her research interests include distributed and parallel systems, middleware, real-time multimedia systems, mobile environments, and formal reasoning of distributed systems. Venkatasubramanian has a PhD in computer science from the University of Illinois at Urbana-Champaign. She is a member of the IEEE and the ACM.

■ Direct questions or comments about this article to Sudeep Pasricha, 444, Computer Science Building, Donald Bren School of Information and Computer Sciences, University of California, Irvine, CA 92697-3425; sudeep@ics.uci.edu.