Quality Adapted Backlight Scaling (QABS) for Video Streaming to Mobile Handheld Devices

Liang Cheng[1],[*], Stefano Bossi[2], Shivajit Mohapatra[1], Magda El Zarki[1], Nalini Venkatasubramanian[1], and Nikil Dutt[1]

[1] Donald Bren School of Information and Computer Science,
University of California, Irvine, CA 92697, USA
{lcheng61, mopy, magda, nalini, dutt}@ics.uci.edu
[2] stboss@tin.it

Abstract. For a typical portable handheld device, the backlight accounts for a significant percentage of the total energy consumption (e.g., around 30% for a Compaq iPAQ 3650). Substantial energy savings can be achieved by dynamically adapting backlight intensity levels on such low-power portable devices. In this paper, we analyze the characteristics of video streaming services and propose an adaptive scheme called Quality Adapted Backlight Scaling (QABS), to achieve backlight energy savings for video playback applications on handheld devices. Specifically, we present a fast algorithm to optimize backlight dimming while keeping the degradation in image quality to a minimum so that the overall service quality is close to a specified threshold. Additionally, we propose two effective techniques to prevent frequent backlight switching, which negatively affects user perception of video. Our initial experimental results indicate that the energy used for backlight is significantly reduced, while the desired quality is satisfied. The proposed algorithms can be realized in real time.

1 Introduction

With the widespread availability of 3G cellular networks, mobile hand-held devices are increasingly being designed to support streaming video content. These devices have stringent power constraints because they use batteries with finite lifetime. On the other hand, multimedia services are known to be very resource intensive and tend to exhaust battery resources quickly. Therefore, conserving power to prolong battery life is an important research problem that needs to be addressed, specifically for video streaming applications on mobile handheld devices.

Most hand-held devices are equipped with a TFT (Thin-Film Transistor) LCD (Liquid Crystal Display). For these devices, the display unit is driven by

---

the illumination of backlight. The backlight consumes a considerable percentage of the total energy usage of the handheld device; it consumes 20%-40% of the total system power (for Compaq iPAQ) [1].

Dynamically dimming the backlight is considered an effective method to save energy [1, 2, 3] with scaling up of the pixel luminance to compensate for the reduced fidelity. The luminance scaling, however, tends to saturate the bright part of the picture, thereby affecting the fidelity of the video quality.

In [2], a dynamic backlight luminance scaling (DLS) scheme is proposed. Based on different scenarios, three compensation strategies are discussed, i.e., brightness compensation, image enhancement, and context processing. However, their calculation of the distortion does not consider the fact that the clipped pixel values do not contribute equally to the quality distortion. In [3], a similar method, named concurrent brightness and contrast scaling (CBCS), is proposed. CBCS aims at conserving power by reducing the backlight illumination while retaining the image fidelity through preservation of the image contrast. Their distortion definition and proposed compensation technique may be good for static image based applications, such as the graphic user interface (GUI) and maps, but might not be suitable for streaming video scenarios, because their contrast compensation further compromises the fidelity of the images. In addition, Neither [2] nor [3] solves the problem associated with frequent backlight switching which can be quite distracting to the end user.

In this paper, we explicitly incorporate video quality into the backlight switching strategy and propose a quality adaptive backlight scaling (QABS) scheme. The backlight dimming affects the brightness of the video. Therefore, we only consider the luminance compensation such that the lost brightness can be restored. The luminance compensation, however, inevitably results in quality distortion. For the video streaming application, the quality is normally defined as the resemblance between the original and processed video. Hence, for the sake of simplicity and without loss of generality, we define the quality distortion function as the mean square error (MSE)(see Equation (1)) and the quality function as the peak signal to noise ratio (PSNR)(see Equation (2)), both of which are well accepted objective video quality measurements.

$$MSE = \frac{1}{M} \times \sum_{i=1}^{M} (x_i - y_i)^2 \tag{1}$$

$$PSNR(dB) = 10 log_{10} \sum_{i=1}^{M} \frac{255^2}{(x_i - y_i)^2} \tag{2}$$

where x_i and y_i are the original pixel value and the reconstructed pixel value, respectively. M is the number of pixels per frame.

It is to be noted that any improved quality metrics may be adopted to replace the MSE/PSNR metrics used here without affecting the validity of our proposed scheme.

As is mentioned in [3], for video applications, the continuous change in the backlight factor will introduce inter-frame brightness distortion to the observer.

In our experiments, we find that the "unnecessary" backlight changes fall into two categories: (1) small continuous changes over adjacent frames; (2) abrupt huge changes over a short period. Therefore, we propose to quantize the calculated backlight to eliminate the small continuous change and use a low-pass digital filter to smooth the abrupt changes.

The rest of the paper is organized as follows. In Section 2, we introduce the principle of the LCD display - experimental results show that backlight dimming saves energy while the pixel luminance compensation results in minimal overhead. In Section 3, we present our QABS scheme, which includes determining the backlight dimming factor and two supplementary methods to avoid excessive backlight switching. Section 4 shows our prototype implementation, experimental methodology and simulation results. We conclude our work in Section 5.

2   Characteristics of LCD

In this section, we outline the characteristics of the LCD unit from two perspectives, the LCD display mechanism and the LCD power consumption, both of which form the basis for our system design.

2.1   LCD Display

The LCD panel does not illuminate itself, but displays by filtering the light source from the back of the LCD panel [2][3]. There are three kinds of TFT LCD panels: transmissive LCD, reflective LCD, and transflective LCD. We focus in this paper on the reflective, since it is the most commonly used LCD for handheld devices. Henceforth, when we mention LCD, we refer to reflective LCD and we refer to both backlight and forelight as backlight. As will be shown, our idea is generic to any backlight based LCD.

The perceptual luminance intensity of the LCD display is determined by two components: backlight brightness and the pixel luminance. The pixel luminance can be adjusted by controlling the light passing through the TFT array substrate. Users may detect a change in the display luminance intensity if either of these two components is adjusted. That is, the backlight brightness and the pixel luminance can compensate each other. In Section 2.2, we will show that the pixel luminance does not have a noticeable impact on the energy consumption, whereas the backlight illumination results in high energy consumption. Hence, in general, dimming backlight level while compensating the pixel luminance is an effective way to conserve battery power in hand-held devices.

Let the backlight brightness level and the pixel luminance value be L and Y, respectively, and the perceived display luminance intensity I. We may denote I using Equation (3).

$$I = \rho \times L \times Y \tag{3}$$

where ρ is a constant ratio, denoting the transmittance attribute of the LCD panel, and as such $\rho \times Y$ is the transmittance of the pixel luminance.
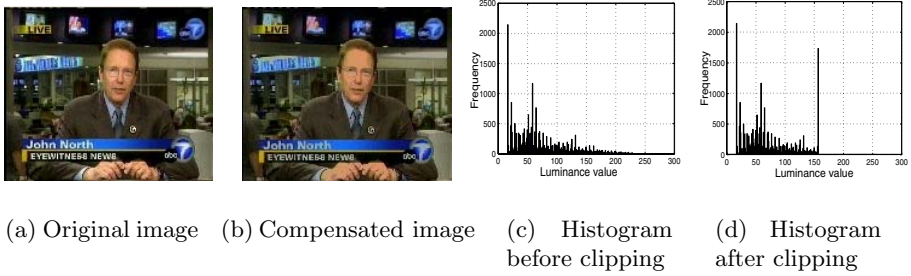
(a) Original image (b) Compensated image   (c) Histogram before clipping   (d) Histogram after clipping

Fig. 1. Image and its luminance histogram before and after clipping

We may reduce the backlight level to L' by multiplying L with a dimming factor α, i.e., $L' = L \times \alpha$, $0 < \alpha < 1$. To maintain the overall display luminance I invariable, we need to boost the luminance of the pixel to Y'. Since the pixel luminance value is normally restricted by the number of bits that represent it (denoted as n), Y' may be clipped if the original value of Y is too high or the α is too low. The compensation of the backlight is described in Equation (4).

$$Y' = \begin{cases} Y/\alpha, & if \ Y < \alpha \times 2^n \\ \\ 2^n, & if \ Y \geq (\alpha \times 2^n) \end{cases}$$

(4)

Combining Equation (4) and Equation (3), we have

$$I' = \begin{cases} I, & if \ Y < \alpha \times 2^n \\ \\ \rho \times L \times \alpha \times 2^n & if \ Y \geq (\alpha \times 2^n) \end{cases}$$

(5)

Equation (5) clearly shows that the perceived display intensity may not be fully recovered, instead, it is clipped to $\rho \times L \times \alpha \times 2^n$ if $Y \geq (\alpha \times 2^n)$. In Figure 2, we illustrate the clipping effect of the display luminance.

In Figure 1-a and Figure 1-c, we show an image and its luminance histogram. This image is the first frame of a typical news video clip ("ABC eye witness news") captured from broadcasting TV signal. Figure 1-b and Figure 1-d illustrate the image and its luminance histogram after backlight dimming and pixel luminance compensation. Figure 1-d shows that the pixels with luminance higher than 156 are all clipped to 156. This clipping effect eliminates the variety in the bright areas, which is subjectively perceived as the luminance saturation and is objectively assessed as 30dB with reference to the original image shown in Figure 1-a.

2.2   LCD Power Model

In our experiments, we observe that the backlight dimming can save energy whereas the compensation process, i.e., scaling up the luminance of the pixel,
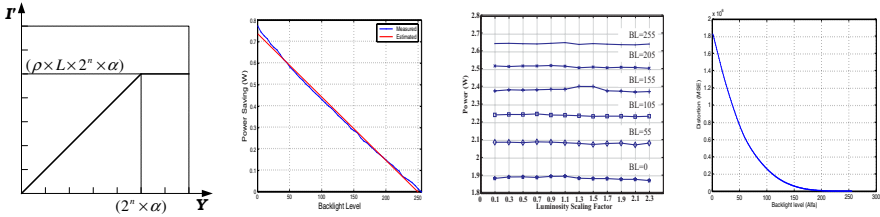
Fig. 2. Clipping   **Fig. 3.** Power saving vs. backlight level   **Fig. 4.** Energy overhead   **Fig. 5.** MSE with different $Alfa$

has a negligible energy overhead. We measure the energy saving as a difference of the total system power consumption with backlight set to different levels from that with the backlight turned to the maximum (brightest). Figure 3 shows the plot between the various backlight levels and their corresponding energy consumption for a Compaq iPAQ 3650 running Linux. A more detailed setup of our experiments is described in Section 4. It is noticed that the backlight energy saving is almost linear to the backlight level and can be estimated using Equation (6).

$$y = a1 \times x + a2 \tag{6}$$

where y is the energy savings in Watt; x denotes the backlight level; a1 and a2 are coefficients. We apply the curve fitting function of MATLAB and obtain $a1 = -0.0029567$ and $a2 = 0.73757$ with the largest residual fitting error as 0.085731.

Contrary to the backlight switching, the pixel luminance scaling is uncorrelated to the energy consumption. In Figure 4, we show that for one specified backlight level (BL) the system energy consumption basically remains stable and is independent of the luminance scaling.

Figure 3 and Figure 4 justify the validity of the generic backlight power conservation approach, i.e., dimming the backlight while enhancing the pixel luminance value. Note that in Figure 4, "BL" refers to the backlight level and "Luminosity Scaling Factor" refers to α. In the next section, we apply this method to the video streaming scenario, discussing a practical scheme to optimize the backlight dimming while taking into consideration the effect on video distortion.

3   Adaptive Backlight Scaling

As explained in Equation (5), the backlight scaling with the luminance compensation may result in quality distortion. The amount of backlight dimming, therefore, has to be restricted such that the video fidelity will not be seriously affected.

3.1   Optimized Backlight Dimming

We define the optimized backlight dimming factor as the one whose induced distortion is closest to a specified threshold. Henceforth, we replace the factor α with the real backlight level $Alfa$, $Alfa = N \times \alpha$ (N is the number of backlight levels (256 for Linux on iPAQ)), and the optimized backlight dimming is represented as $Alfa^*$.

In Figure 5, we illustrate the image quality distortion in terms of MSE over different backlight levels. (Note that we use the image shown in Figure 1-a.) We see that as $Alfa$ increases, the induced video quality distortion due to the brightness saturation monotonously decreases. Hence, for a given distortion threshold, we can find a unique $Alfa(= Alfa^*)$ for each image. In video applications, for a given distortion, different frames may have distinct $Alfa^*$, depending on the luminance histogram of that frame. However, it is hard to have an accurate analytical representation of the quality distortion using $Alfa$ as a parameter. We therefore adopt an optimized search based approach, where we calculate the MSE distortion with different $Alfa$ until the specified distortion threshold is met. The results of our scheme are accurate and can be used as the benchmark for the design of other analytical methods.

Figure 6 shows the exhaustive searching algorithm for finding $Alfa^*$ for one image. $FindAlfa(th)$ takes the distortion threshold (th) as input, and returns the $Alfa^*$ as output. Note that $MSE(Alfa)$ calculates the MSE with the specified $Alfa$ for one frame.

However, the complexity of an exhaustive search shown in Figure 6 is too high. As shown in Equation (2), the per-frame MSE calculation consists of M multiplications and $2M$ additions. M is the number of pixels in one frame, e.g., $M = 25344$ for QCIF format video. We regard the per-frame MSE as the basic complexity measurement unit. We assume that the optimized backlight level is uniformly distributed in [0, N], and thus the complexity of algorithm in Figure 6 is $O(N)$. In our test, $N = 256$. Obviously, the optimized backlight dimming factor can hardly be calculated in real-time.

Therefore, we apply a faster bisection method [4] to improve the algorithm for finding $Alfa^*$. Since we can easily find an upper bound (denoted as u) and a lower bound (denoted as d) on the backlight levels, we get as good an approximation as we want by using bisection. We assume that $u > d$ and let ϵ be the desired precision and present the algorithm in Figure 7.

By using the bisection method, we may achieve the complexity of $O(log_2 N)$ in the worst case. For instance, for $N = 256$ and $\epsilon = 1$, we only need to calculate per-frame MSE at most eight times, which is fast enough for real-time processing.

3.2   Smoothing the Backlight Switching

It has been discussed in [3] that the backlight dimming factor may change significantly across consecutive frames for most video applications. The frequent switching of the backlight may introduce an inter-frame brightness distortion to the observer. Hence, it is necessary to reduce frequent backlight switching.

Proc: FastFindAlfa(th, ϵ)
1: u := upper bound;
2: d := lower bound;
3: **while** (u - d) $>$ ϵ) **do**
4:     Alfa = round((d + u)/2);
5:     **if** (MSE(Alfa) $>$ th) **then**
6:        u = Alfa;
7:     **else**
8:        d = Alfa;
9:     **end if**
10: **end while**
11: return(Alfa);

Proc: FindAlfa(th)
1: Alfa := 0;
2: **while** Alfa $\leq$ N **do**
3:     **if** MSE(Alfa) $>$ th **then**
4:        Alfa := Alfa + 1;
5:     **else**
6:        return(Alfa);
7:     **end if**
8: **end while**

Fig. 6. Exhaustive algorithm for finding Alfa*

Fig. 7. Fast algorithm for finding Alfa*

In our study, we observe that the calculated $Alfa^*$, although based on an individual image, does not experience huge fluctuations during a video scene, i.e., a group of frames that are characterized with similar content. Actually, the redundancy among adjacent frames constitutes the major difference between the video and the static image application and has long been utilized to achieve higher compression efficiency. Hence, the backlight switching should be smoothed out within the scene and most favorably only happen at the boundary of video scenes.

We propose two supplementary methods to smooth the acquired $Alfa^*$ in the same video scene. First, we apply a low-pass digital filter to eliminate any abrupt backlight switching that is caused by the unexpected sharp luminance change. The passband frequency is determined by the subjective perception of the "flicker moment" and the frame display rate. Second, we propose to quantize the number of backlight levels, i.e., any backlight level between two quantization values can be quantized to the closest level, by which we prevent the needless backlight switching for small luminance fluctuations during one scene. In our experiments, we quantize all 256 levels to "N"levels (N=5 in our study). We switch the backlight level only if the calculated $Alfa^*$ changes drastically enough, so that it falls into another quantized level.

4   Performance Evaluation

In this section, we introduce our prototype implementation, the methodology of our measurement and the performance of the proposed algorithm.

4.1   Prototype Implementation

Figure 8 shows a high level representation of our prototype system. Our implementation of the video streaming system consists of a video server, a proxy server

and a mobile client. We assume that all communication between the server and the mobile client is routed through a proxy server typically located in proximity to the client.

The video server is responsible for streaming compressed video to the client; The proxy server transcodes the received stream, adds the appropriate control information, and relays the newly formed stream to the mobile client (Compaq iPAQ 3650 in our case). For the sake of simplicity and without loss of generality, in our initial prototype implementation, we use the proxy server to also double up as our video server.

The proxy server includes four primary components - the video transcoder, the proposed QABS module, the signal multiplexer, and the communication manager. The transcoder uncompresses the original video stream and provides the pixel luminance information to the QABS module. The QABS module calculates the optimized backlight dimming factor based on the user quality preference feedback received from the client (user). The multiplexer is used to multiplex the optimized backlight dimming information with the video stream. The communication manager is used to send this aggregated stream to the client.

On the mobile client, the demultiplexer is used to recover the original video stream and the encoded backlight information from the received stream. The LCD control module renders the decoded image onto the LCD display. The backlight information is fed to the "Backlight Adjustment Module", which concurrently sets the backlight value for the LCD. In particular, users may send the quality request to the proxy when requesting for the video, based on his/her quality preference as well as concern for battery consumption.

4.2   Measurement Methodology

For video quality and power measurements, we use the setup shown in Figure 9. The proxy in our experiments is a Linux desktop with a 1GHz processor and 512MB of RAM. All our measurements are made on a Compaq iPAQ 3650. We use a National Instruments PCI DAQ board to sample voltage drops across a resistor and the iPAQ, and sample the voltage at 200K samples/sec. We calculate the instantaneous and average power consumption of the iPAQ using the formula $P_{iPAQ} = \frac{V_R}{R} \times V_{iPAQ}$.
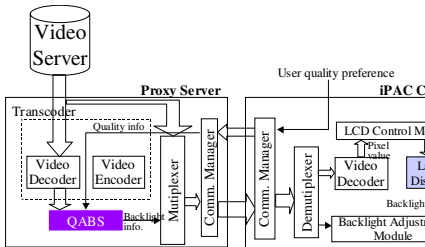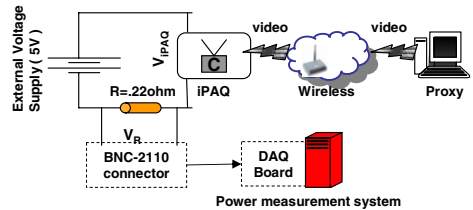

Fig. 8. Prototype implementation

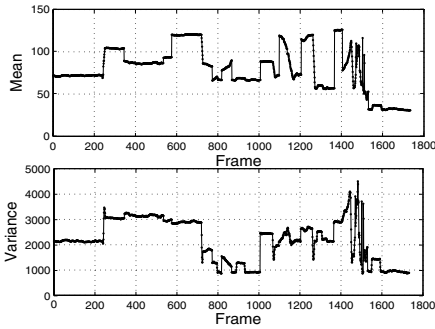**Fig. 9.** Setup for our measurements

Fig. 10. Basic statistics of *abc_news*
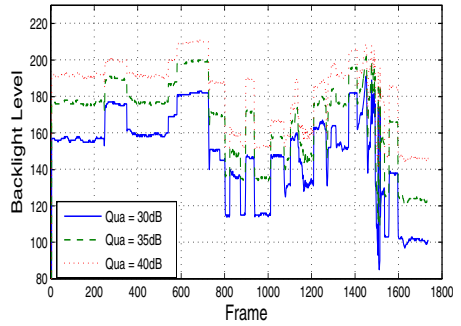
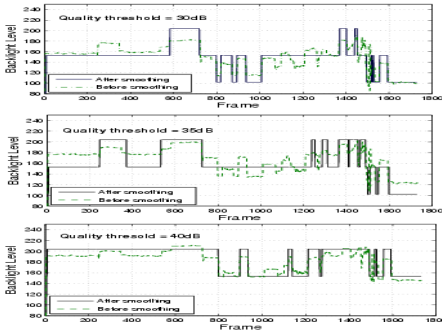**Fig. 11.** *Alfa** adapted to three given quality thresholds

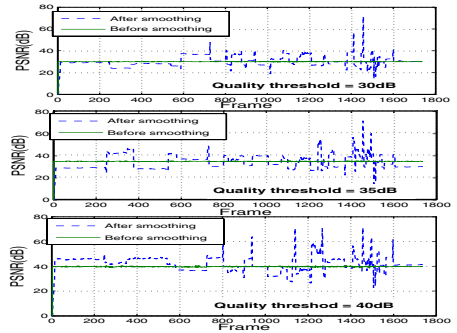**Fig. 12.** *Alfa** before and after filtering and quantization

Fig. 13. Quality before and after Alfa smoothing

4.3   Experimental Results

In our simulation, we use a video sequence captured from a broadcasted *ABC_news* program, whose first frame is shown in Figure 1-a. We choose this video as representative of a typical usage of a PDA. In Figure 10, we show the basic statistics (i.e., the mean and the variance of luminance per frame) of this video.

We assume that the users are given three quality options, fair, good, and excellent, which respectively correspond to the PSNR value of 30dB, 35dB, and 40dB. After applying the algorithm "Proc: FastFindAlfa", we obtain the adapted *Alfa** for these three quality preferences, as is shown in Figure 11. It can be seen that higher video quality needs higher backlight level on average.

In Figure 12, we show *Alfa** before and after the backlight smoothing process. It is seen that the small variation and the abrupt change of the backlight switching are significantly eliminated after the filtering and quantization. In addition, as we expected, the backlight switching mostly happens at the boundary of major scenes.

In Table 1, we summarize the results of our QABS. The mean *Alfa** of different quality preferences produces a quality on average very close to the

Table 1. Results of QABS

Alfa Mean			Quality(dB)			Power Saving(%)		
Fair	Good	Excellent	Fair	Good	Excellent	Fair	Good	Excellent
149	162	186	30.17	34.28	42.31	41.8%	36.7%	27.3%

pre-determined quality threshold. It is noted that different quality requirements result in various power saving gains. Higher quality preference must be traded using more backlight energy. Nevertheless, we can still save 29% energy that is supposed to be consumed by the backlight unit if we set the quality preference to be "Excellent".

In Figure 13, we show that the filtering and quantization process may lead to instantaneous quality fluctuation, which is contrasted to the consistent quality before backlight smoothing. Nevertheless, we observe that the quality fluctuation is around the designated quality threshold and mostly happens at scene changes.

5   Conclusion

In this paper, we apply a backlight scaling technique to video streaming applications, and explicitly associate backlight switching to the perceptual video quality in terms of PSNR. The proposed adaptive algorithm is fast and effective for reducing the energy consumption while maintaining the designated video quality. To reduce the frequency of backlight switching, we propose two supplementary schemes that smooth the backlight switch process such that the user perception of the video stream can be substantially improved.

Acknowledgement

References

1. S. Pasricha, M. Luthra, S. Mohapatra, N. Dutt, N. Venkatasubramanian, "Dynamic Backlight Adaptation for Low Power Handheld Devices," *To appear in IEEE Design and Test (IEEE D&T), Special Issue on Embedded Systems for Real Time Embedded Systems,* Sep. 2004.
2. N. Chang, I. Choi, and H. Shim, "DLS: Dynamic Backlight Luminance Scaling of Liquid Crystal Display," *IEEE Transaction on VLSI System,* vol. 1, Aug. 2004.
3. W.-C. Cheng, Y. Hou, and M. Pedram, "Power Minimization in a Backlit TFT-LCD Display by Concurrent Brightness and Contrast Scaling," *Proceedings of the Design,* Automation and Test in Europe, Feb. 2004.
4. J. L. Zachary, *Introduction to Scientific Programming: Computational Problem Solving Using Maple and C.* Telos Publishers, 1996.