Adaptive Display Power Management for Mobile Games

Bhojan Anand‡, Karthik Thirugnanam†, Jeena Sebastian‡, Pravein G. Kannan‡
Akhihebbal L. Ananda‡, Mun Choon Chan‡ and Rajesh Krishna Balan†
‡School of Computing, National University of Singapore
†School of Information Systems, Singapore Management University
‡{banand,jeena,pravein,ananda,chanmc}@comp.nus.edu.sg, †{karthikt,rajesh}@smu.edu.sg

ABSTRACT

In this paper, we show how tone mapping techniques can be used to dynamically increase the image brightness, thus allowing the LCD backlight levels to be reduced. This saves significant power as the majority of the LCD's display power is consumed by its backlight. The Gamma function (or equivalent) can be efficiently implemented in smartphones with minimal resource cost. We describe how we overcame the Gamma function's non-linear nature by using adaptive thresholds to apply different Gamma values to images with differing brightness levels. These adaptive thresholds allow us to save significant amounts of power while preserving the image quality. We implemented our solution on a laptop and two Android smartphones. Finally, we present measured analytical results for two different games (Quake III and Planeshift), and user study results (using Quake III and 60 participants) that shows that we can save up to 68% of the display power without significantly affecting the perceived gameplay quality.

Categories and Subject Descriptors

K.8.0 [**General**]: Games; I.4.3 [**Image Processing & Computer Vision**]: Enhancement; I.3.3 [**Computer Graphics**]: Picture / Image Generation—*Display Algorithms*

General Terms

Algorithms, Design, Human Factors, Experimentation

Keywords

Power Management, Tone Mapping, Backlight Scaling, Mobile Games

1. INTRODUCTION

The current generation of mobile smartphones are not just devices for voice communication. Instead, they are frequently used as

mobile PC replacements that can edit documents, browse the web, check email, and play games. To enable these rich applications, smartphones are now equipped with reasonably fast CPUs, fairly large amounts of memory, and fast network connections. However, this power comes at the cost of battery lifetimes. Indeed, to maintain the slim form factor required for these phones (which impacts the amount of battery that can be put into the phone), it is quite common for the battery lifetimes of smartphones to be significantly shorter compared to the previous generation of "dumber" phones.

In modern smartphones, the three main sources of power consumption are, 1) the CPU, 2) the display, and 3) the network interfaces. We found that the wireless interface and display components dominate the power consumption.

In particular, on a HTC Magic Android smartphone, with all components running at peak levels, the LCD display and 3G network interface consumes 45 to 50% and 35 to 40% of the total system power respectively. The remaining power is consumed by the CPU and memory subsystems. Unfortunately, games, which account for more than 50% of current iPhone application downloads [32], tend to consume large amounts of CPU, memory, and display power in general — with multiplayer games also consuming large amounts of network interface power.

In this paper, we present our system for reducing the power consumption of mobile multiplayer games. We focus primarily on the display interface and defer to previous work for the CPU [20, 21] and network components [1, 2, 3] respectively. The common display power saving technique of dimming the display whenever possible is difficult to apply to highly interactive games as it is difficult to find situations where the user is not looking at the display. In addition, non-gradual display changes will be very user noticeable, as jitter etc., due to the high frame rates of games (20–80 frames per second). Our solution uses the following observations.

1. For any given image, it is possible to achieve comparable perceived brightness in multiple ways. In the normal case, the brightness of the image and the display are set to default levels. However, we can also achieve similar perceived brightness by brightening the image or content and dimming the display, or vice versa, by darkening the image and brightening the display.

2. There are two options for brightening the image - linear and non-linear. Linear approaches increases all the pixels in an image by the same value while non-linear approaches such as tone mapping [23] apply different values to different pixels. Non-linear approaches, unlike linear approaches, do not suffer from high pixel saturation (where parts of the image look excessively bright) and are already available for use in may 3D game rendering engines. We discuss our non-linear approach using tone mapping further in Section 2.2.1.

3. The amount of perceived image distortion varies with different brightness levels even when the same tone mapping function is applied. We explain in Section 2.2.1 how we factor in perceived quality to our non-linear tone mapping mechanisms to ensure that game images are never brightened to the point where details start washing out.

With these three insights, we devised a solution that involves brightening all the frames in a game using non-linear tone mapping techniques followed by dimming the brightness of the display. This results in significant power savings for LCD technology-based displays as the extra CPU power needed to brighten the frames is more than offset by the power saved by dimming the display. In addition, users are unable to notice any quality differences between the brightened and original images. We call this approach *Adaptive Backlight Management* and present its algorithm, implementation in two real games, and evaluation results in the rest of this paper.

To test our approach, we used the Quake III [22] first person shooting (FPS) game and the Planeshift [5] massively multiplayer role playing game (MMORPG). We chose these two games as 1) the code for both games have been open sourced, 2) both are popular games; Quake III is a commercial multi-player game, with a very sophisticated code base, that has sold millions of copies, and Planeshift has been in operation since 2006 and has a large loyal player-base, and 3) it is difficult to save display power in FPS and MMORPG games, as they involve large amounts of real-time player interaction (players are constantly trying to shoot or interact with each other). In addition, these types of games tend to be among the most popular types of games played and downloaded [32]. As such, we are confident that if our approach can work well with these two games, it will also work quite well with other mobile and PC games.

We evaluated our approach using a combination of analytical and user testing. First, we measured the display power saved using two different LCD displays — a 15 inch laptop display and two 3.2 inch smartphone displays (HTC Magic & HTC Hero). In addition, we conducted a user study, involving 60 undergraduates, to measure the perceived quality of our solution. Our results show that we are able to save up to 47% of the display power with no perceptible quality loss and up to 68% of the display power if the user is willing to tolerate a small amount of quality loss. Overall, the main contributions of this paper are as follows:

1. We present a general and systematic approach to save energy that first dynamically adjusts the brightness of images using a widely available tone mapping function and then reduces the LCD display's backlight levels.

2. We formulate the acceptable tone mapping thresholds, across various content brightness levels, that can save significant amounts of power while preserving the end-user gameplay experience.

3. Our approach has been implemented, using open-sourced commercial games, on both laptops and Android Mobile phones.

4. We show that power savings can be achieved even in games with high frame rates that feature fast and near-constant player interaction.

2. BACKGROUND

Our solution is optimised for TFT LCD technology-based displays. We discuss in Section 5.1 how our solution performs on other display technologies such as OLED etc. In the rest of this
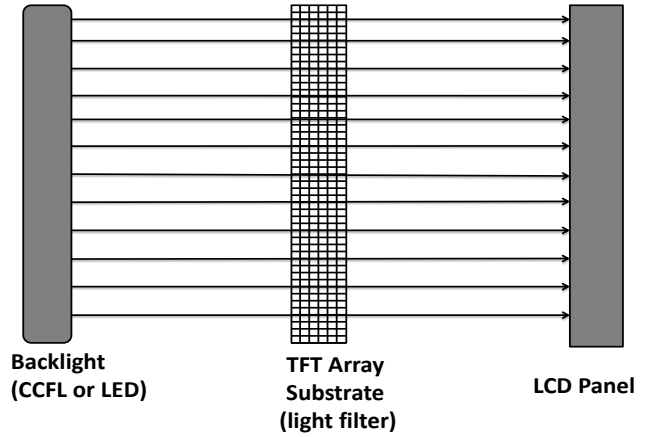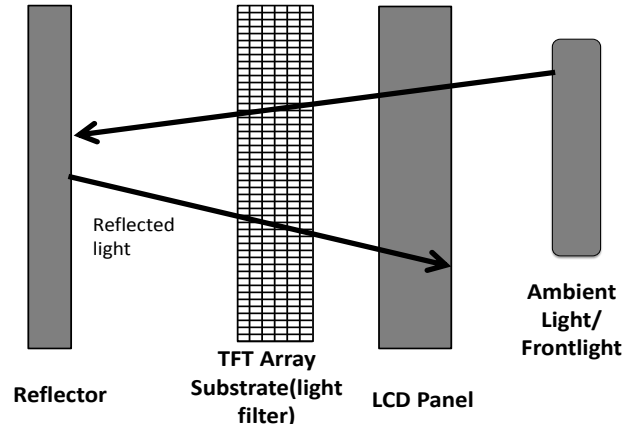

Figure 1: Transmissive LCD Displays

Figure 2: Reflective LCD Displays

section, we first review the characteristics of LCD technology and identify the backlight / frontlight components as the major sources of power consumption. Next, we present the concept of *Tone Mapping* and describe its advantages over linear pixel-by-pixel transformation methods.

2.1 LCD Displays

TFT LCD displays have two major components: the LCD panel and a light source, called the backlight, at the back of the LCD panel that illuminates the panel [11, 13]. The panel filters the backlight based on the values of the pixels in the display buffer — the brighter the pixel, the more light from the backlight is allowed through. Traditional LCD displays use Cold Cathode Fluorescent Lamps (CCFL) for the backlight while modern displays used LED arrays instead.

There are three kinds of TFT LCD panels: transmissive, reflective and transflective [12] . The transmissive displays (Figure 1) use the backlight to illuminate the pixels and offer a wide colour gamut and high contrast. Transmissive displays provide the best visual quality under a broad range of lighting conditions — from completely dark to office lighting for example. However, they tend to be unsuitable for extremely bright environments where the ambient light overpowers the backlight.

Reflective LCDs (Figure 2) are unique because they operate without the use of a backlight. Instead, they rely solely on ambient light. This allows the device to consume far less power than transmissive display. Reflective displays provide best performance in bright outdoor areas, but require a frontlight to improve the viewing ex-

perience in dimly lit environments. A transflective LCD uses both transmissive and reflective methods. It uses a backlight, similar to transmissive displays, but also adds a reflective mirror. Transflective LCDs are a compromise that allows good performance under any lighting condition.

In general, all the three types need a light source, either backlight or frontlight to illuminate the display. Almost all the power in an LCD display is consumed by the backlight or frontlight, with the filter itself requiring comparatively low power. Hence, reducing the backlight or frontlight power consumption is beneficial to all these three types of displays. In the following sections we primarily focus on reducing the power consumption of the backlight. Our methods are equally applicable for displays with frontlights.

2.2 Tone Mapping

As stated earlier, a transmissive LCD display consists of a backlight that shines through a filter (TFT array substrate). This filter is controlled by the pixel values that are to be displayed and modulated by the backlight. Our solution takes advantage of the fact that the perceived brightness is due to the backlight being modulated by the filter, which in turn is controlled by the frame content to be displayed. A dark frame appears dark because the filter does not allow much of the light to shine through. However, the same effect can be achieved by using a dimmer backlight, and controlling the filter to allow more light to shine through. That is, the backlight can be dimmed by boosting the frame content to achieve the same level of perceived brightness (PB). PB is the final brightness output and is what the user finally sees on the LCD screen.

To increase the brightness of the image, the pixel values need to be increased for all pixels in the frame [30, 11, 13]. This can be done in two ways; linearly and non-linearly. There are two limitations with using a linear approach; First, saturation of pixel values will result in poor content quality (the image will look overly bright in some areas). Second, doing pixel-by-pixel transformations for every frame is computationally expensive on mobile devices and could result in less energy savings (due to high CPU costs) and lower frame rates [23] — making it impractical for games that demand high frame rates.

Hence, we instead used a non-linear *Tone Mapping* technique to increase the brightness of the image. In general, tone mapping is a technique used in image processing and computer graphics to map one set of colours to another. One of the common use of tone mapping is to compress the images with high dynamic range (HDR) to a lower dynamic range (LDR) so that they can be displayed or printed on devices which have limited dynamic range. Dynamic range is defined as the range of light intensities present in a scene, which is generally high in real world images. More details about tone mapping is available elsewhere [7, 8, 9, 27]. In this paper, we use tone mapping to change the brightness levels of an image while preserving the image contrast.

Tone mapping operators are usually classified as either global (spatially uniform) or local (spatially varying). Global operators apply a single luminance transform function to every pixel in the image while local operators apply non-linear transform functions to selected pixels. As stated above, these linear global operators are simple and computationally efficient, but have difficulty effectively preserving local contrast in most HDR images. Local operators solve this problem by using a non-linear spatially varying mapping — two identical input luminances may be mapped to different output values based on properties of their local neighbourhood. As shown in Section 2.2.1, a local non-linear operator achieves much better quality than a linear global operator.

Tone mapping operators can be efficiently implemented at hard-

ware level [23] with specified quality constraints using the function defined below.

Tone Mapping Function. Let I be the Perceived display Brightness of the image, α be the set of pixels of the image, BL be the Backlight Level and D_{max} be the maximum tolerable Image Distortion. The objective of tone mapping function is to find a mapping operator for a given I that maps $\alpha \rightarrow \beta$ such that, BL is minimised with the distortion $\leq D_{max}$.

2.2.1 Using the Gamma Function for Tone Mapping

In this paper, to lower computational overheads on mobile devices, we use a standard Tone Mapping operator called the *Inverse Gamma Function*. (γ). This function is also called *Gamma Compensation* or *Gamma Correction* and is widely available in all graphics implementations (OpenGL, X11, game rendering engines, etc.). In this paper, we simply refer to it as the **Gamma** function. Gamma's non-linearity slows down saturation heavily; thus allowing increased energy savings with minimum quality loss.

The Gamma function is already implemented in many commercial 3D games, including Quake III, and we leverage this function to change the brightness of the displayed image. In 3D games, some of the rendered objects are supposed to represent illumination objects such as lights etc. This is achieved by applying Gamma. In addition, to make game objects appear photo-realistic, Gamma is always applied during the final rendering phase [4]. Since Gamma is already being applied by 3D games in the final rendering phase, we can perform our image brightness changes, using Gamma, without incurring any additional computational cost. In addition, Gamma is also available as a hardware-assisted function (reducing the computational cost even further) and is already a part of many modern mobile GPUs such as the Nvidia Tegra 2 [29].

The effect of changing the brightness of an image using both a linear approach and the non-linear Gamma-based approach is shown in Figure 3. The three images shown are the original image, the linearly brightened image and the Gamma brightened image. The linearly brightened and Gamma brightened images save the same amount of power. We also provide the histograms and colour saturation values for each image. *Colour saturation* is defined as the percentage of pixels where any one of the pixel's sub components (Red, Green or Blue in RGB colour space) have been clipped (greater than 255 in 8-bit representation) due to transformations.

We observe that both the linear approach and Gamma shift the image histogram to the right (indicating increased brightness). However, Gamma retains the shape of the original histogram while the linear approach exhibits high distortion (due to high pixel colour saturation). We observed that increasing the Gamma value increases the mid-tones of an image by a large amount with a comparatively smaller increase for the extreme black and white pixels. This non-linear property makes it hard to determine, for any given image, how much we can dim the backlight to achieve the same overall perceived brightness. We explain in Section 3.2 how we used high precision light sensors to obtain a good relationship between Gamma and perceived brightness.

Changing the Gamma of an image also changes its global contrast as shown in Figure 4. We define *global contrast* as the standard deviation among all the pixel values in the image. A low contrast results in the image appearing "washed out" as all the pixels look similar. In particular, applying a higher Gamma value results in a higher contrast loss. Thus, we cannot brighten the image too much if we want to preserve the image quality.

In addition, we also observed that the rate at which an image loses contrast varies according to the current brightness and colour of the image. Bright or light coloured images lose contrast faster
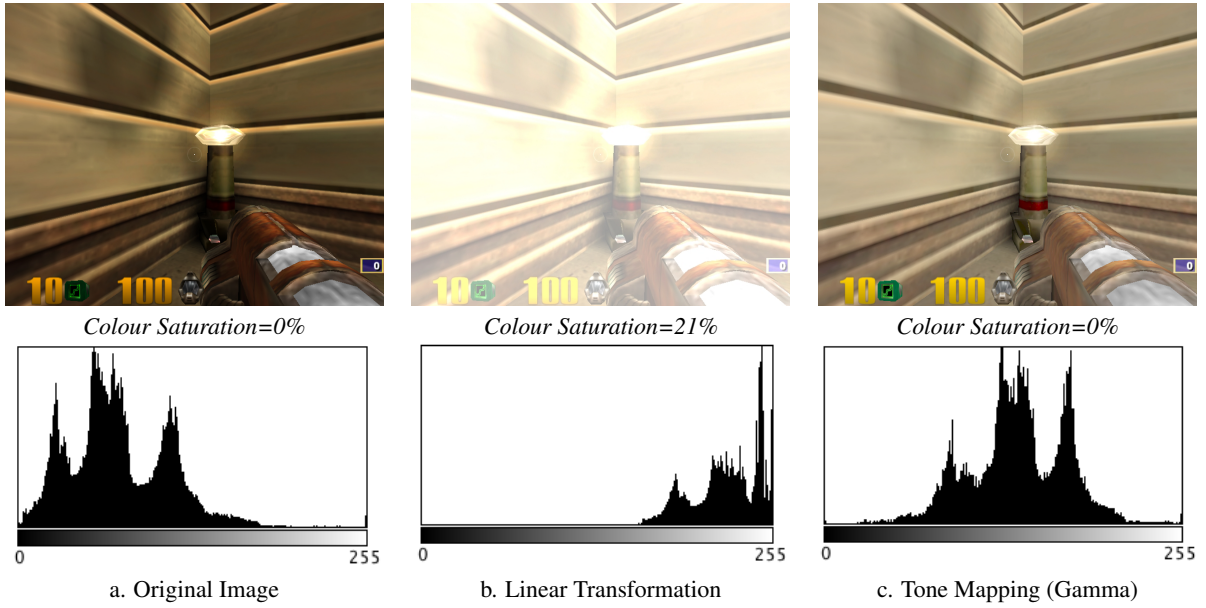
Colour Saturation=0%   *Colour Saturation=21%*   *Colour Saturation=0%*

a. Original Image   b. Linear Transformation   c. Tone Mapping (Gamma)

Figure 3: The Effect of Gamma and Linear Transformations. The Amount of Power Saved is the Same for Both Approaches

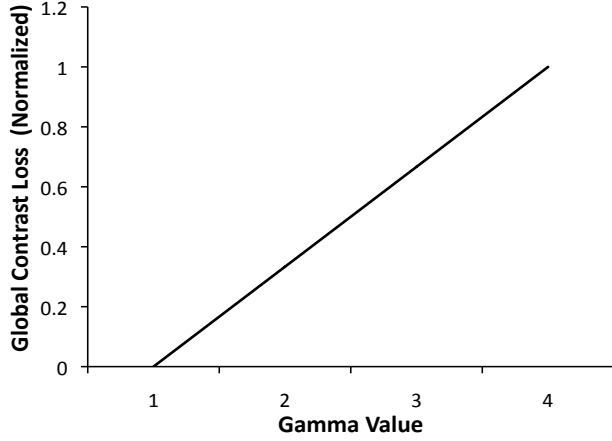**Figure 4: Global Contrast Loss vs Gamma**

Figure 5: Global Contrast Loss vs Image Brightness for $\gamma = 3$

than dark images. This is shown in Figure 5 where the change in contrast for ten images (image brightness level 1 is the darkest image and image brightness level 10 is the brightest image) is shown. We applied a constant gamma value of 3 to all ten images shown in the Figure. Hence, even for the same Gamma value, different images will exhibit different levels of quality (contrast) loss.

To account for this variable contrast loss, we dynamically change the amount of Gamma correction applied according to the brightness level of the current game environment. If the game environment is bright (outdoors, full lighting, etc.), we apply a smaller gamma boost compared to dark game environments. We show in Section 3.5 how we used a user study to obtain the maximum Gamma that can be applied to different brightness levels while still retaining the quality (contrast) at acceptable levels.

3. SYSTEM DESIGN

The basic design of our system is as follows; We understand that most of the LCD display power is consumed by the display's backlight. This backlight-approach is inefficient as most of the energy is wasted when displaying a darker image. To reduce this inefficiency, we reduce the backlight level of the TFT LCD screen (conserving
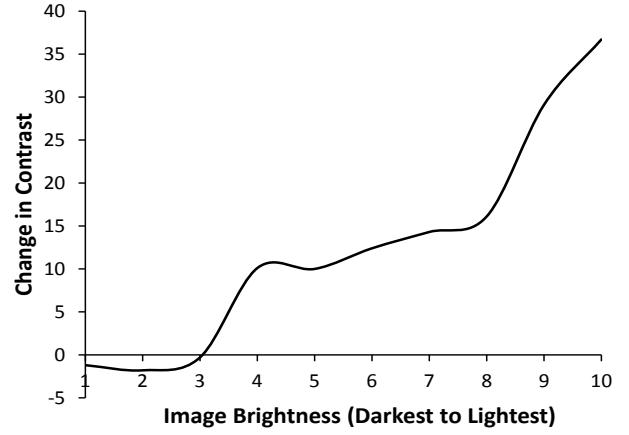
energy), and compensate for this reduction with content enhancement methods in order to maintain (to the best of our ability) the intended quality of the image being displayed.

As described in Section 2.2.1, we use the Gamma function to boost the image brightness. By using Gamma, which is already used in the frame rendering pipeline, we do not add any significant computational overhead. However, the non-linear nature of Gamma (affects the contrast of images in non-linear was) prevents us from just applying Gamma uniformly to all images. This naive approach would result in significant and varying user perceived quality loss.

In the rest of this section, we describe how to operationalise this approach by first studying the relationship between the backlight intensity, image brightness, and the power consumed. We then conduct an experiment to calculate the maximum possible reduction in backlight that can be achieved, by using different values of Gamma, that still preserves perceived image quality. We then describe how to calculate the amount of compensation (in terms of backlight reduction) needed when changing Gamma for a particular image. Finally we describe our complete run time adaptive system that dynamically adjusts Gamma and the backlight levels to
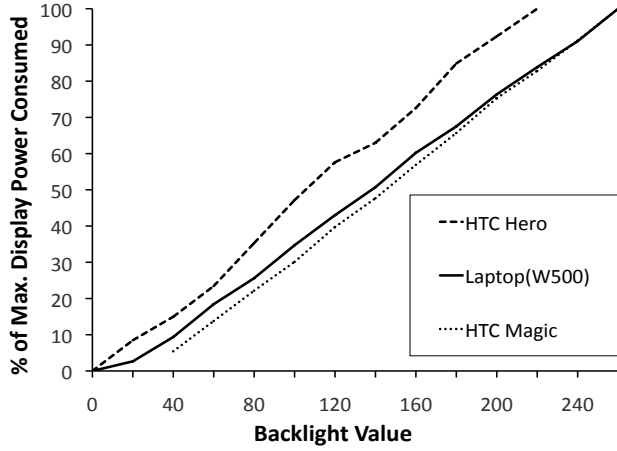
Figure 6: Power vs Backlight level

Figure 7: Backlight level vs Gamma value

achieve significant power savings while preserving user perceived image quality.

3.1 Backlight Power Measurement

First, we measured the amount of power consumed by various backlight levels. We first measured the average current drawn by a 15 inch Lenovo W500 laptop with different backlight brightness levels (with negligible CPU/network load throughout the experiment) using the measurement system shown in Figure 12. We then repeated the experiment on the HTC Magic & HTC Hero smartphones, using the Power Tutor [35] software to measure the the backlight power values on each smartphone.

Our measurements, shown in Figure 6, showed a linear relationship between the backlight level and the percentage of power consumed by the backlight. It is thus possible to save significant amounts of energy by reducing the backlight level of LCD screens.

3.2 Gamma to Backlight Relationship

We now describe our methodology to determine the mapping between the backlight reduction and the Gamma-induced associated image compensation. As Gamma is a non-linear and complex function, we studied the relationship between Gamma, image brightness, and backlight levels through a series of experiments. These experiments were all conducted in a dark room using precision light sensors from Phidgets [31] which can measure human perceptible light levels ranging from 1 lux (deep twilight) all the way to 1000 lux (Overcast day). The aim of the experiment was to determine the appropriate amount of Gamma that was needed to compensate for backlight reduction under different conditions.

We determined this as follows; We first set the Gamma and backlight intensity to their default values (1 Gamma and 255 (full) backlight) and measured the perceived brightness (PB) of the screen using a light sensor placed close to the screen and pointing at it. Then, we increased the gamma by Δ and found the appropriate backlight level (BL) such that the PB was constant and similar to the base level. Equation 3.1 shows this functional relationship.

$$BL = f(\gamma) \; : \; PB \; is \; constant \qquad (3.1)$$

For this experiment we used sample images from the Quake III game. The value of the Gamma parameter ranged from 1 to 10, where 1 indicates default frame content and higher values result in brighter content. The backlight levels ranged from 0 to 255, where 0 indicates the backlight was off and 255 was the brightest backlight level.
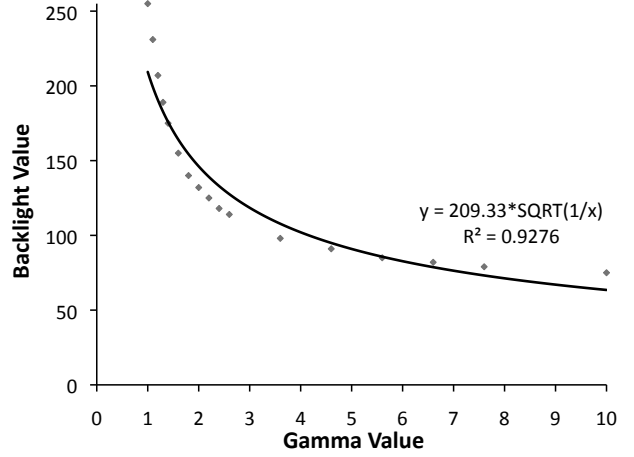
3.2.1 Analysis of Gamma to Backlight Relationship

Figure 7 shows the results of this experiment for various Gamma levels. We fitted an exponential curve on the plotted data and tested its fitness quality using R^2 regression analysis. R^2, called the *coefficient of determination* is a commonly used statistic to check the *goodness of fit*. R^2 ranges from 0 to 1 with a value close to 1 indicating a good fit. The curve in Figure 7 has an R^2 value of 0.9276, indicating an excellent fit.

From this curve, we obtained Equation 3.2 (generated from the functional relationship between BL and gamma shown in Equation 3.1) and used it in our algorithms to map from gamma values (γ) to the backlight level (BL).

$$BL = 209.33 \sqrt[2]{1/\gamma} \qquad (3.2)$$

The equation shows the non-linear nature of Gamma as lower values of Gamma change the brightness levels of the image significantly more than higher values of Gamma. As our goal is to increase the image brightness (and thus decrease backlight levels), we use lower values of Gamma in our algorithm. In particular, small values of Gamma result in a big difference in the content brightness.

3.3 Measuring Image Quality

A key question in our approach is to understand the impact on image quality that our use of Gamma with backlight reduction introduces. From Figure 3, we have some evidence that Gamma does not impose as significant quality loss. But how do we measure this objectively?

To answer this question, we use a variety of simple and widely accepted [12] objective quality metrics used by the image processing community. We use *Mean Square Error* (MSE) and *Peak Signal to Noise Ratio* (PSNR) as our two quality metrics. MSE and PSNR compare the modified image with the original image to give the deviation in terms of the distortion and quality gain respectively.

Let x be the pixel intensity array of the original image and y be the pixel intensity array of the modified image. The MSE between these two images is, the mean value of $(x_i - y_i)^2$ for all pixels in x and y. This is shown in Equation 3.3. Higher MSE means higher distortion in the modified image.

$$MSE(x,y) = \frac{1}{N} x \sum_{i=0}^{N} (x_i - y_i)^2 \qquad (3.3)$$

PSNR is derived based on MSE as shown in Equation 3.4. The

| Original | (MSE=0.05, PSNR=+12.91dB) | (MSE = 0.01, PSNR = +22.84 dB) |
| - | (MSSIM=+0.79) | (MSSIM=+0.94) |

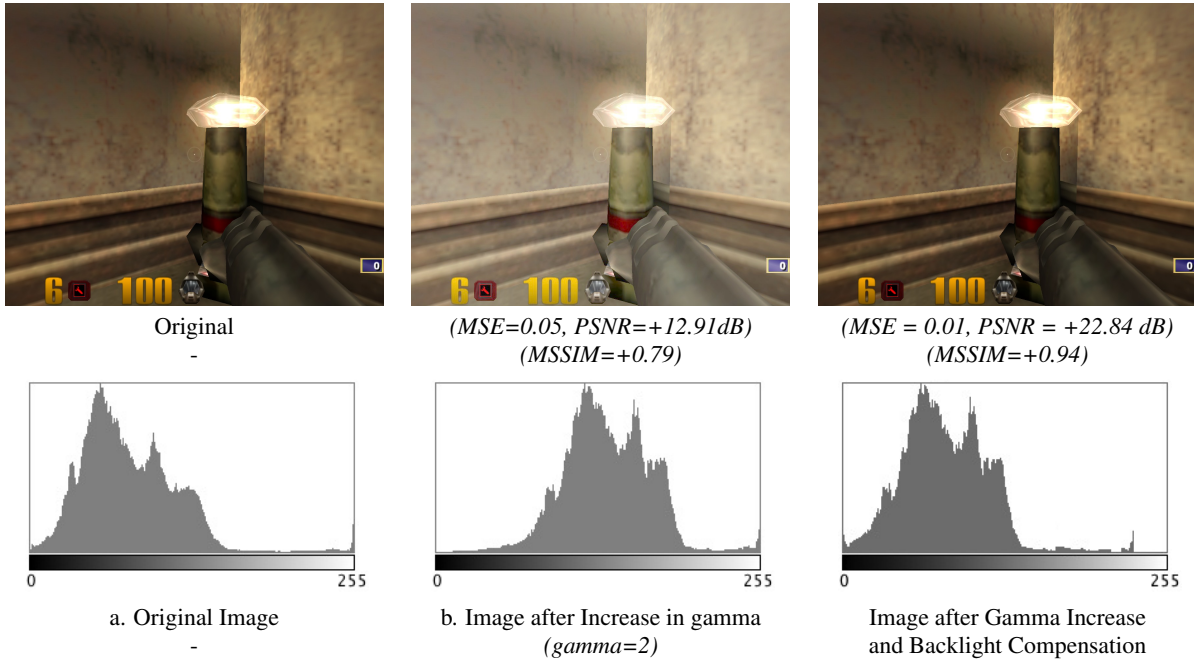| a. Original Image | b. Image after Increase in gamma | Image after Gamma Increase |
| - | (gamma=2) | and Backlight Compensation |

Figure 8: Effect of Gamma Increase and Compensation

MAX value indicates the dynamic range of a pixel. For example, images with 8-bits per pixel will have $2^{8-1} = 255$ as the dynamic range. In our computations we have normalised all the pixel values to the range 0 to 1 and hence, the dynamic range is 1 ($MAX = 1$). Higher PSNR means the quality of the modified images is comparable to the original image.

$$PSNR(dB) = 20log_{10}(\frac{MAX}{\sqrt[2]{MSE}}) \qquad (3.4)$$

Though MSE and PSNR are widely used for their simplicity, they are merely objective measurements and they ignore the attributes of *Human Visual Perception* (HVS). To account for HVS, we also use a more complex metric, called *Structural SIMilarity Index* (SSIM), which accounts for human perception and is gaining increasingly popularity among the image processing community.

The principle philosophy underlying the SSIM approach is that HVS is highly adapted to extracting structural information from visual scenes. In particular, images are highly structured and these structures are important cues about the perceived quality of an image. To account for this structural property, SSIM breaks the image into patches and compares the quality of each patch with other patches using three attributes of quality; namely luminance, contrast and structural similarity.

Let *x* be the pixel array of the patch from the original image and *y* be the pixel array of the patch from the modified image. The SSIM Index between these two patches, SSIM(x,y) is defined in Equation 3.5. The index ranges from -1 to +1, with -1 representing high distortion and +1 representing low distortion (high quality). The SSIM index will be +1 when a patch is compared with itself.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \qquad (3.5)$$

where,

- μ_x the average of *x*;
- μ_y the average of *y*;
- σ_x^2 the variance of *x*;

- σ_y^2 the variance of *y*;
- σ_{xy} the covariance of *x* and *y*;
- $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilise the division with weak denominator;
- *L* the dynamic range of the pixel-values (typically, this is $2^{no. of bits per pixel} - 1$);
- $k_1 = 0.01$ and $k_2 = 0.03$ by default. These values are obtained through experiments [36].

In this paper, we are interested in comparing two complete images. Hence we use the *Mean SSIM* (MSSIM) value, shown in Equation 3.6, that averages the SSIM values of all the patch comparisons. A more detailed explanation of the SSIM, including its complex mathematical derivation and improvements over existing approaches is available elsewhere [36].

$$MSSIM(x,y) = \frac{1}{N}\sum_{j=1}^{N} SSIM(x_j, y_j) \qquad (3.6)$$

where *N* is the number of patches or windows.

3.3.1 Using the Image Quality Metrics

Figure 8 shows an image after applying a Gamma level of two to it (Figure 8b) and after decreasing the backlight to compensate (Figure 8c). We observe that the histogram of the final backlight-compensated image is closer to the original image compared to the image that only had Gamma applied. In addition, the final image has a MSSIM score of +0.94 which is very close to a perfect +1 and better than the middle image's score of +0.79. The PNSR value is also in the 20 dB to 25 dB range which has been reported to be of high quality on mobile devices [25, 34].

Overall, the images tell us that it is possible to achieve power savings with little quality loss and that we need to adjust the backlight not only to save power but also to improve the image quality.

3.4 Computing Image Brightness

As described above, our approach changes the brightness of an image using Gamma. However, first we have to measure the current brightness of the image to understand how much Gamma to apply. A naive approach would be to measure the brightness of each image pixel and then compute an average value. However, this is computationally intensive and unnecessary.

We instead estimate the image brightness through careful sampling. To do this, we select $\approx$2000 samples pixels (1 out of every 20 pixels) from the image. For each pixel, we compute its brightness as a function of the pixel's Red (R), Blue (B), and Green (G) colour values. This is a common calculation, shown in Equation 3.7, used in image processing. In general, brightness values range from 0 to 255, where lower values represent darker images and higher values represent brighter or lighter images.

$$^1 PixelBrightness = \sqrt[2]{.241xR^2 + .691xG^2 + .068xB^2} \qquad (3.7)$$

We then compute a weighted average (favouring pixels towards the centre of the screen) of all the pixel brightness as the final image brightness value. This average value is then discretized into 14 levels to allow for efficient use in the rest of our computations. These 14 levels are not linear, and were selected to provide as much detail as possible for the common lighting values seen in our test games, while still covering the entire brightness range. We achieved this by using brightness levels 1 to 12 to cover the entire expected brightness ranges and reserved levels 0 and 13 to indicate values that are too dark and too bright respectively.

3.5 Human Calibration of Gamma Thresholds

3.5.1 Objective

To maximise the amount of energy conserved, we need to know the maximum amount of Gamma that can be applied to a particular image such that the resultant perceived image quality, after backlight compensation, is still acceptable to the game player. As Gamma is non-linear, the same amount of Gamma increase results in different levels of quality loss for different images. If the image is already too bright, increasing Gamma will result in a "washing-out" of the image and there will be a significant amount of loss in overall contrast. Compensation by dimming the backlight will not restore the lost contrast. We thus conducted a short user study to study the acceptable values of Gamma for each image brightness level.

3.5.2 Methodology and Setup

For this user study, we used 4 non-author Masters and 6 non-author PhD students from the same lab as the authors. Seven of the ten participants were Male and they had an average age of 26 years. To run this study, we designed a gamma slider tool, which allowed the users to slide the gamma value to an acceptable level. We used Equation 3.2 to automatically adjust the backlight to appropriate levels when the participants changed the Gamma value using the tool.

We used the above-mentioned 14 discrete brightness levels, and selected 2 different images for each brightness level (28 images in total) from the Quake III game. The goal of the user study was to obtain an *aggressive threshold* for each image. The aggressive threshold is the highest Gamma level that might show quality loss but not up to the level where the images are washed-out.

_____

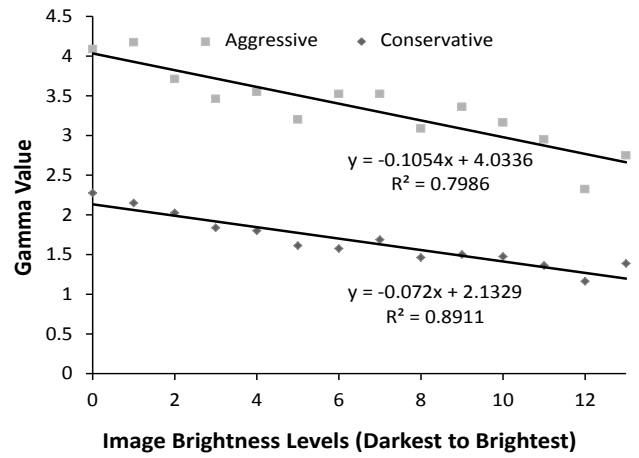[1](http://alienryderflex.com/hsp.html)

Figure 9: User-Perceived Acceptable Gamma Levels

We used the same experiment to also obtain the *conservative threshold* for each image. The conservative threshold is the highest gamma level which still results in images of comparable quality to the original image. This threshold is always lower than the aggressive threshold and preserves quality at the expense of power savings.

3.5.3 Results

With 10 users and 2 images per brightness level, we had a total of 20 points for each of the gamma thresholds. These points were averaged to yield the final threshold values for each image brightness level. These points are shown in Figure 9 for each brightness level. The image brightness levels of 0 to 13 correspond to the darkest to brightest images available in the game.

The figure shows a definite trend and that for different image brightness levels, users prefer different Gamma compensation values. In particular, as image brightness increases, the threshold values decrease. The trend lines gives $R^2 = 0.7986$ *and* $R^2 = 0.8911$, which indicate good quality fits for both the trend lines. From the trend we get Equations 3.8 and 3.9 for adjusting Gamma. Equation 3.8 corresponds to a set of aggressive values which gives the highest power saving with acceptable image and gameplay quality. Equation 3.9 corresponds to set of conservative values which gives the best image and gameplay quality with significant amounts of power saving.

$$\begin{aligned} Aggressive: \quad & \gamma = -0.10 \, x \, CBL + 4.03; \\ & where \, CBL \, = \, Content \, Brightness \, Level \end{aligned} \qquad (3.8)$$

$$\begin{aligned} Conservative: \quad & \gamma = -0.07 \, x \, CBL + 2.13; \\ & where \, CBL \, = \, Content \, Brightness \, Level \end{aligned} \qquad (3.9)$$

3.6 Objective Analysis of Gamma Thresholds

In addition to obtaining user-derived Gamma thresholds for various image brightness levels, we also analysed the correlation between our user study results and common objective metrics. We used *global contrast change (loss)* (GCC) as the metric to measure image quality, as it is one of most accepted quality metrics in the image processing industry [18], and quite suitable for measuring the quality loss due to Gamma correction. GCC uses the standard deviation among all the pixels in an image to compute the global
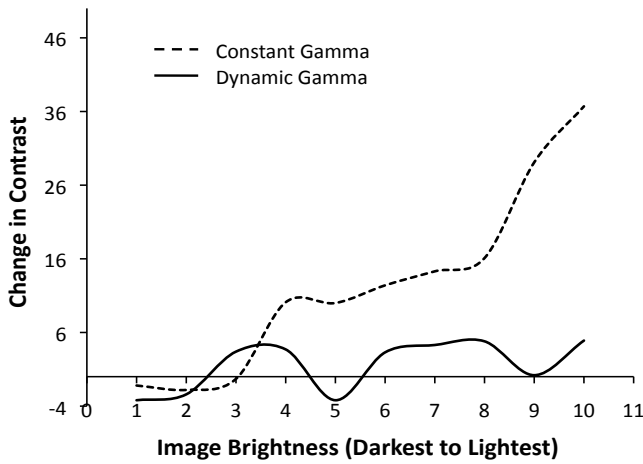
Figure 10: Contrast Loss vs Adaptive Gamma

contrast of an image. A change in this standard deviation indicates contrast error.

Our analysis of the conservative threshold (Equation 3.9) shows a minimal contrast change (less than 4 on average, on a scale of (-100 to +100)). Similarly our aggressive threshold (Equation 3.8) keeps the contrast change within 16 on average.

Figure 10 shows the effect on contrast, using the conservative thresholds, on images with different brightness levels (1 is the darkest, 10 is the brightest). This figure, when compared to the earlier Figure 5 (also shown in Figure 10 as the dotted line), clearly demonstrates the importance of using different Gamma values for different image brightness levels. In particular, our adaptive thresholds do not the suffer the same high contrast losses seen in the previous approach where a constant Gamma value was applied. We quantify the amount of power saved and the effect on user-perceived quality of both the aggressive and conservative thresholds in Sections 4.1 and 4.2.

3.7 Run-time Algorithm

Putting all the insights gained in the earlier parts of this section together, the actual run time portion of our final system, shown in Figure 11, can be described as follows:

- Every 250ms, our algorithm computes the brightness value of the current image (or frame). We run our algorithm every 250ms only as brightness levels in a game tend to remain constant for this amount of time (players can't move faster than this). We then compute an average brightness level using the brightness values of current image and the four previous images.

- If the average brightness value is the same as the previous average brightness value, the display settings are left unchanged and the algorithm finishes the run.

- If the brightness value is different, a new gamma value is calculated using either the conservative (Equation 3.8) or aggressive (Equation 3.8) thresholds according to the game player's preferences (save as much power as possible without quality loss or save even more power with some quality loss). The backlight level is then adjusted to match the Gamma value using Equation 3.2.

We use an average brightness value (of the current image and the previous four images) to determine if we need to change the Gamma and backlight levels. This was intentionally done to ensure
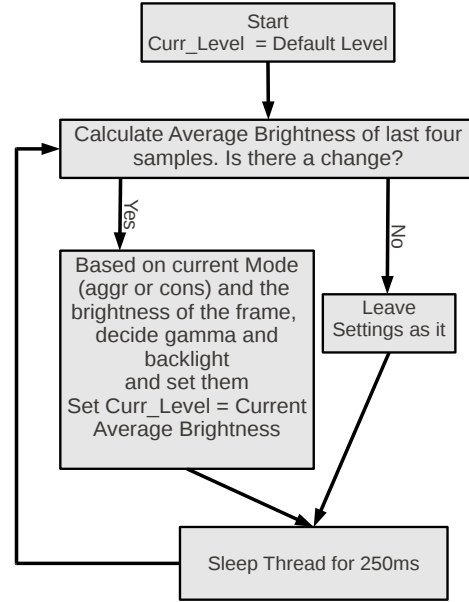

Figure 11: Flowchart of our System

that increases and decreases of brightness happens gradually, usually one level at a time. This reduces the probability of big changes in backlight values which cause highly noticeable flickering effects.

4. EVALUATION & RESULTS

In this section, we present our evaluation methodology and the results of the evaluation of our system. Our goal was to find the amount of power saved by our system and its impact on the quality of the gameplay.

To calculate the amount of power saved we measured the power consumed by the original game and modified versions of the game that use our system. These results are presented in Section 4.1. Due to the complexity in modelling human visual perception, we could not evaluate the gameplay quality analytically. Instead, we performed an extensive user study to evaluate the quality of modified versions of the games that use our system. The user study methodology, setup and results are described in Section 4.2.

4.1 Analytical Evaluation

4.1.1 Measurement Setup

We performed our power measurements on a Lenovo W500 laptop and two different Android smartphones (HTC Magic & HTC Hero). The measurements were also done with two different games, Quake III and Planeshift. Different variants of Quake III and Planeshift, that implemented different versions of our system, were played throughout the measurements and the power consumed by each of the variants was recorded.

We used five different versions of the game: *static-conservative*, *dynamic-conservative* (the conservative thresholds shown in Equation 3.9), *static-aggressive*, *dynamic-aggressive* (the aggressive thresholds shown in Equation 3.8) and the *default* (no power-saving) version. Static-conservative used the lowest Gamma value among all the dynamic-conservative threshold values and *static-aggressive* used the highest Gamma value among all the dynamic-
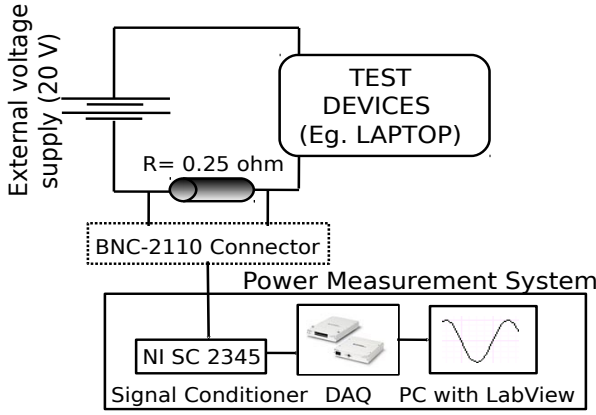
Figure 12: Laptop Power Measurement Setup

aggressive threshold values. Thus the static cases represent and test the extreme cases of our system.

We used the difference between the original version and each of the power optimised versions to calculate the power saved by each version. To ensure that our power measurements were repeatable, reliable, and could be usually compared between the different versions, we used the same pre-recorded gameplay sessions with each game version.

We measured the power consumed on the laptop by first removing the battery and then intercepting the power intake. The current consumed (in Amps) was measured using a National Instruments Data Acquisition (DAQ) device, as shown in Figure 12, for a period of over one minute with the voltage set at 20V. The total energy consumed was then calculated using the voltage and current values. For the smartphones, we used the Power Tutor application [35] that measured the power consumed by the mobile display over a period of time in Joules.

4.1.2 Baseline Measurements

State			Energy consumed over 1 minute (Joules)	Total Energy consumed by LCD Display (Joules)	% Energy consumed by LCD Display
CPU	Network	Display			
MIN	OFF	BLANK	1936.62		
MIN	OFF	FULL	2353.30	416.56	17.70
MIN	FULL	BLANK	2167.69		
MIN	FULL	FULL	2541.52	372.82	14.70
FULL	OFF	BLANK	2741.95		
FULL	OFF	FULL	3177.27	435.31	13.70
FULL	FULL	BLANK	2889.86		
FULL	FULL	FULL	3307.32	417.46	12.62

Table 1: Baseline Power Consumption of the Laptop

To provide a baseline for our system, we first measured the regular power consumption of both the laptop and the smartphones under different scenarios. These baseline measurements are shown in Table 1 for the laptop and Table 2 for the HTC Magic (the HTC Hero was similar).

For the laptop, the power consumption of the LCD screen varied

from 12.6% (with the CPU and network busy) to 17.7% (with the CPU and network idle). Note that the LCD power consumption is relatively small as the W500 laptop uses an Intel Core2 Duo T9600 CPU that draws significantly more power than a smartphone processor. In addition, background activities, and other components such as the RAM and HDD also consume significant amounts of energy.

State			Energy over 1 minute (Joules)			Total Energy Consumed (Joules)	% Energy Consumed By LCD Display
CPU	Network (3G & WiFi)	Display	CPU	Network	Display		
MIN	OFF	FULL	1.5	0	46.1	47.6	96.84
MIN	FULL	FULL	1.5	41	46	88.5	51.97
FULL	OFF	FULL	12	2	47	61	77.04
FULL	FULL	FULL	15	41	46	102	45.09

Table 2: Baseline Power Consumption of the HTC Magic

For the HTC Magic, we observe that the LCD Display, as expected, consumes a far greater percentage of the total system power. In particular, the display consumes between 45% (when the CPU and network are busy) to 96% (when the CPU and network are idle) of the entire system power consumption.

Game Variant	Quake III (%)	Planeshift (%)
Dynamic-aggr	68.26	67.07
Static-aggr	69.79	68.40
Dynamic-cons	48.95	47.89
Static-cons	20.04	21.73

Table 3: Power-Savings Measurements

4.1.3 Measured Analytical Results

In this section, we provide the power saving achieved by our system. Table 3 shows the results for Quake III and Planeshift. These values are the power savings achieved on the laptop. We verified, using the mobile version of Quake III called kwaak3 [24], that the power savings for Quake III were similar on the mobile phone. We were unable to run Planeshift on the Android smartphones as there was no useful mobile port of the game.

Overall, the power savings for both games were similar and allow us to draw some broader conclusions. In particular, the static-conservative variant saves the least amount of display power (about 21%) but it also has the lowest quality loss. However, the dynamic-conservative variant is able to save $\approx$47% of the display power while achieving similar quality (as shown by the user study in Section 4.2).

If the user is willing to tolerate some amount of quality loss, the dynamic-aggressive variant saves the same amount of display power ($\approx$68%) as the static-aggressive variant, but with significantly better perceived quality (Section 4.2).

4.2 User Study & Analysis

In the previous section, we showed that we can save between 21% to 68% of the display power. However, how does this affect the perceived user quality? In this section, we describe our user study that investigates the impact of our system (and all its variants) on user perceived quality.

4.2.1 Methodology, Participants, and Setup

The user study was performed using Quake III and three different game maps that spanned the entire brightness levels. The maps used were: a darker map (q3tourney4), a brighter map (simpsons-q3) and a map with the combination of darker and brighter areas (reqbath).

We recruited a total of 60 undergraduate students, from Singapore Management University (SMU), for our study. Participation was open to all students at SMU and we solicited participation through university emails. Our participants were a mix of students from technical and non-technical majors. We used Lenovo Thinkpad laptops (Model T61/T60/W500) with external mice for the user study to avoid input modality issues (playing Quake III on a smartphone is quite hard).

Each user study took about 60 minutes to complete. Before testing our system, each participant completed a short (2-3 minutes) demographics survey to determine their familiarity with networked games. The questions were:

1. Frequency of computer/mobile gaming;

2. Experience with computer/mobile games in general;

3. Experience with FPS (First Person Shooting) game;

4. Experience with Quake III game.

Table 4 summarises the participant demographics.

The study protocol was as follows: each participant was first given a short training session to teach them how to play the default version (no power savings) of Quake III on a non user-study map. After the training session, each participant was asked to play all 4 power saving variants of the game on each of the three maps. The order of maps and power saving variants were randomised for every participant. We also included the default non-power saving variants as one of the random experiments to test if the participants were able to differentiate the highest quality non-power saving baseline from the other 4 variants.

Thus, in total, each participant played 5 Quake III games of 3 minutes each for each of the three maps. To avoid player confusion, we made each participant finish all 5 experiments for a particular map before moving to a completely new map (although we did randomise the order in which they played the maps).

For each map, the participants were asked to play the baseline non-power saving variant before playing each of the other 5 versions. This was to allow them to accurately calibrate the quality of each variant to the best possible quality for that map. As mentioned above, for each map, the 5 variants were presented in random order to avoid learning and ordering effects.

After testing each variant, each participant was provided a simple questionnaire with six questions (shown below) that they had to rate using a 5-point Likert scale marked with the adjectives "Strongly-Agree", "Somewhat-Agree", "Neutral", "Somewhat-Disagree" and "Strongly-Disagree". The questions were:

1. The game WAS as playable as the baseline.

2. The colours of objects WERE differentiable and clear;

3. All the objects in the game WERE clearly viewable;

4. The darker and brighter areas in the game WERE distinguishable and clear;

5. The shadows or light-rays WERE clearly viewable;

6. The cross-hair pointer of your weapon WAS visible and clear in all areas of the game.

Total Number	60
Gender	Male (34), Female(26)
Proficiency level in computer/mobile games	Novice (16), Average (37), Expert (6), Never played computer/mobile games(1)
The frequency of playing computer/mobile games	Almost every day (22), Few times in a week (18), Few times in a month(17), I never play games (3)
Played any FPS game before	Yes (43), No (11), Not sure about the game type(6)
Played Quake game before	Yes (8), No (33), Have not heard about Quake game(19)

Table 4: Demographics Statistics for the User Study

4.2.2 User Study Results

The results of the user study, shown in Figure 13, can be summarised as follows:

- For all 6 questions, for both static and dynamic settings, the users are able to consistently differentiate the high quality (low Gamma) and lower quality (high Gamma) variants.

- As expected, the user rating for static-conservative is very close to the default non-power saving variant. However, dynamic-conservative has comparable user ratings even though it achieves significantly more power savings (47% versus 21%) than the static-conservative variant.

- The user rating of dynamic-aggressive is noticeably better than static-aggressive even though both save about the same amount of power (68%).

- The results for all 3 maps are consistent.

To check for outliers, we calculated the standard deviation among all our samples and found that they were at acceptable levels. The maximum standard deviation was 1.14 and the average standard deviation was 1.07.

4.3 Overall Result: System Works Very Well

We now combine the results from both the analytical power measurements (Section 4.1) and the user study (Section 4.2) to understand the trade-offs involved in saving LCD display power for mobile games. The first trade-off is that when high quality is required, it is clear that the dynamic-conservative scheme is the best variant as it saves significant amounts of display power (47%) while achieving comparable perceived quality to the more conservative static-conservative variant and the default full-quality non-power saving variant.
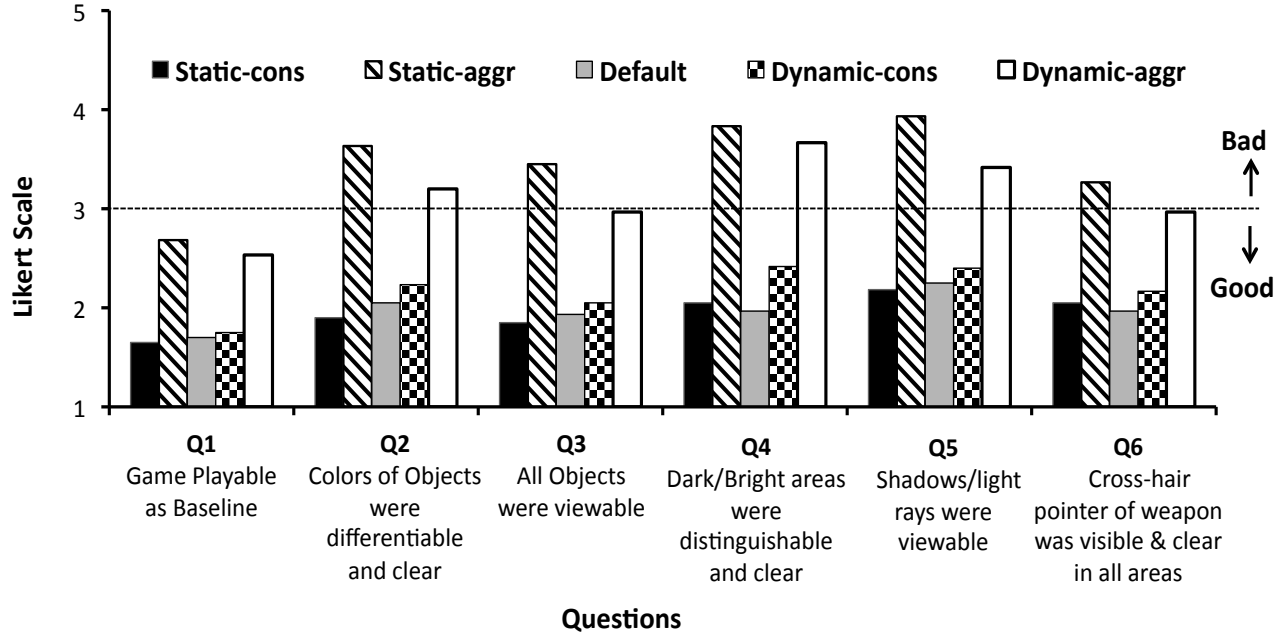
All values are average scores across all 3 maps. Pair-wise two-tailed t-test results were as follows: There were no significant differences between the Default and Static-cons and Dynamic-cons values. All the values for Static-aggr (for all 3 maps) were significant (at 5%) against the Default values. For Dynamic-aggr, all values for maps 2 and 3 (light map and light/dark map) were significant at 5%. Q4 and Q5 for map 1 (dark map) were significant at 5% with Q1, Q2, Q3, and Q6 being non-significant.

Figure 13: Results of the User Study for all 3 Maps. All Versions of the Game were Deemed Playable by the Participants.

However, when energy efficiency is more important and the user is willing to trade-off quality for energy, then both the dynamic and static algorithms with high gamma settings can save about 68% of the display power. In terms of quality, the dynamic-aggressive variant achieves consistently better user ratings than static-aggressive and should thus be the variant that is used when higher power savings is desired. In particular, even though users noticed quality degradation with the dynamic-aggressive variant, they still consistently rated the game as being highly playable on all 3 maps.

In summary, our adaptive (dynamic) approach, using either conservative or aggressive settings, saves substantial amount of display energy (47% to 68%) and offers excellent quality versus energy trade-offs.

5. DISCUSSION

5.1 Applicability to OLED Displays

The system presented in this paper is tuned specifically for TFT LCD display. However, there are other types of displays that are used for smartphones. In particular, organic light-emitting diode (OLED) is an emerging display technology that is becoming increasingly popular and has been used in smartphones such as the Google Nexus One (Active Matrix OLED or AMOLED, a variant of OLED) and the Samsung Galaxy S (Super Active Matrix OLED or SMOLED, another similar variant of OLED). An OLED display uses light-emitting diodes (LED), embedded in an emissive electroluminescent layer made from a film of organic compounds, that emit light in response to an electric current.

The biggest difference between a TFT LCD and an OLED display is that an OLED display functions without a backlight as each LED in an OLED display is lit individually. OLED displays consume less power (about 40% less) than a LCD display when the image is dark, but consumes more power than a LCD display (up to 3x more under certain conditions) when the image is bright (as
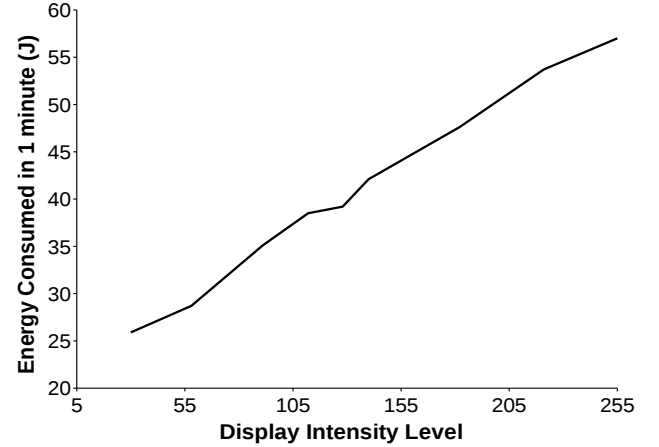

Figure 14: OLED Energy Consumption vs Screen Brightness

more LEDs need to be lit) [15]. We confirmed this trend with our own power measurements on the Google Nexus One smartphone with a 3.7 inch AMOLED display.

To perform this experiment, we kept the displayed image constant and varied the brightness of the display while measuring the power consumption of the display. Figure 14 shows the results of this experiment. As expected, the power consumption of the display varied linearly to the display brightness (255 is the maximum brightness). This is similar to LCD displays.

However, Figure 15 shows the power consumption of the display when we kept the brightness of the display constant but applied different Gamma values (from 1.0 to 2.0) to the displayed image. Again, we found a linear relationship. This is quite different from LCD displays where the power consumption is determined almost solely by the brightness level of the backlight (and not on the brightness of the image that has been increased using Gamma).
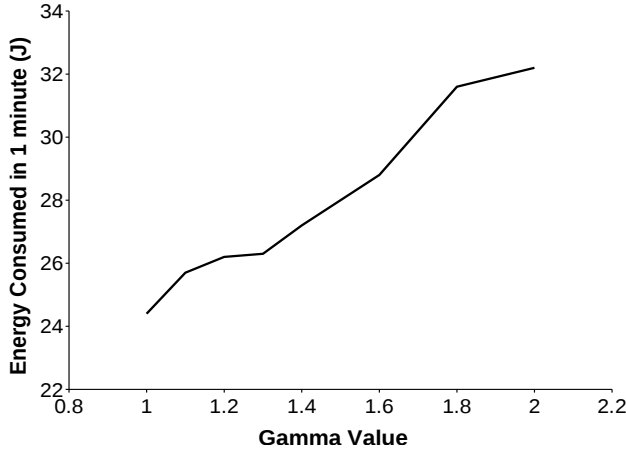
Figure 15: OLED Energy Consumption vs Gamma Value

This result suggests that to save power on OLED displays, we need to *darken* the image instead of brightening it as brighter images consume more power. This can be done by reducing the Gamma value instead of increasing it. However, what effect does reducing Gamma have on the image quality? To test this, we decreased Gamma from 1 to 0 in gradual increments and measure the contrast change of the resulting image. Figure 16 shows the results of this experiment. We observe that decreasing Gamma has a gradual yet increasing effect on the contrast.

The core objective of the LCD-optimised adaptive display power management algorithm presented earlier is to keep the contrast change within acceptable levels for frames with different brightness. We can use the same adaptive mechanism to save energy for OLED displays — but by darkening (instead of brightening) the image appropriately. We are currently investigating this approach and determining the correct Gamma thresholds to achieve high power savings with minimal quality loss.
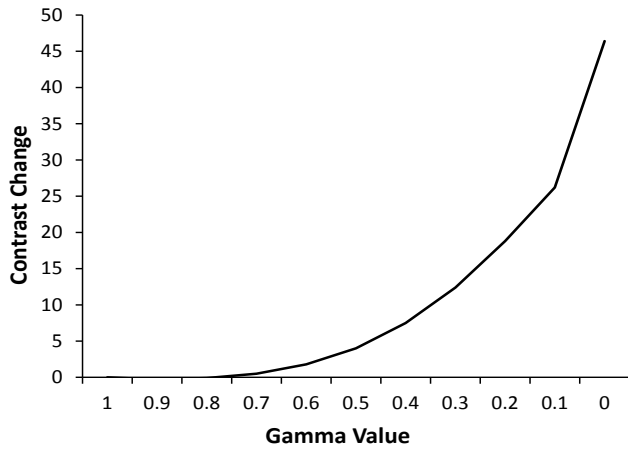

Figure 16: Change in Contrast vs Gamma

Finally, we observed that the energy consumption of OLED displays is quite sensitive to the color being displayed. To demonstrate this, we measured the base energy consumption of the Nexus One's OLED display for a period of 1 minute with the red, green, and blue colour intensities all set to 0 (i.e., we displayed a completely black image. This is base power consumption reference point).

Next, we gradually changed only the red color intensity (with green and blue intensities both set to 0) and measured the power consumption of the red display components at each intensity level.

We then repeated this experiment for just the blue and green colours. After each experiment, we subtracted the power measurements from the base power consumption (black image) to get the incremental power consumption caused by that colour and intensity.

The results, shown in Figure 17, show that red consumes the least energy with green consuming approximately 1.5 times more energy than red, and blue consuming approximately 2.1 times more energy than red. In addition, we also discovered that it was possible to predict the power consumption of any image by adding the individual power consumptions of the red, green, and blue pixels that collectively made up that image. We intend to incorporate this unique colour property of OLED displays into our adaptive algorithm and we hope to report our results in the near future.
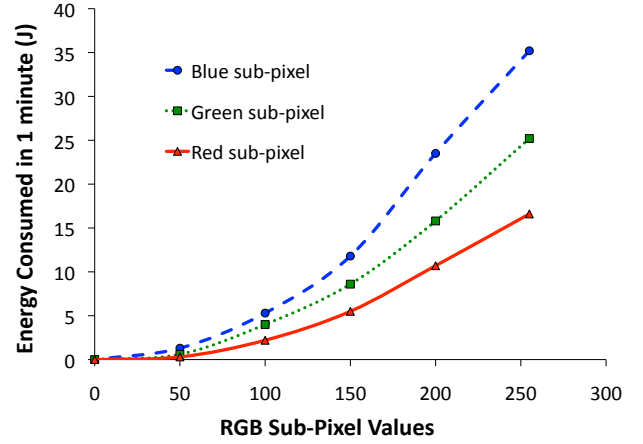

Figure 17: Energy Vs RGB Sub-Pixel Values

5.2 Lack of Hardware and Software γ support

Implementing our system on laptops is easy as most laptop graphics cards come with hardware Gamma support. In addition, Linux distributions for laptops (Ubuntu, Redhat, Debian etc.) also come with software support for manipulating Gamma. However, this was not true for Android smartphone. We found it impossible to change the Gamma value directly in Android 2.2 and below smartphones as we did not have the appropriate software interfaces and the graphics hardware did not support direct Gamma manipulation.

Hence, for our Android smartphone implementation, we did not use Gamma directly, but instead used the Alpha Blending techniques of the OpenGL library to efficiently mimic the required Gamma manipulations in the rendering loop of the game's graphics pipeline. We implemented this solution in the kwaak3 [24] Android port of the Quake III game and verified that the solution (in terms of expected power savings and resulting image quality) was similar to the Gamma-based approach used on the laptop. A visual demonstration of our implementation running on a HTC Desire HD Android phone (we did not use this more powerful phone to obtain results for this paper), is downloadable from our server [28].

However, moving forward, the latest smartphones, such as the Samsung Galaxy S II, come with dual core CPUs and powerful GPU co-processors (e.g, the Nvidia Tegra 2 [29]) that have hardware Gamma support. Hence, our solution will become even easier to implement on the next generation of smartphones.

5.3 Limitations and Future Work

Our user studies are a key component in designing and evaluating the quality of our system. However, they do have limitations. Our first user study that found the gamma thresholds was done with only a limited number of postgraduate computer science students.

A general or larger population might show slightly different results. Also, this study was done in a well controlled office lighting environment. A similar study should be done in various lighting conditions such as indoor and outdoor environments to generalise our algorithm. The second user study that evaluated the gameplay quality was also conducted in a controlled office lighting environment using just one class of participants (undergraduates).

Our Android smartphone power measurements may not be highly accurate or fine grained as they were obtained using a third-party software application. We may need to use more accurate measurement techniques on additional phone models and screen types. Our solution uses the default brightness level of the image — factoring in other parameters, such as the default contrast level and default colour level, could make our dynamic adaptation more accurate. Finally, contrast loss in images can be corrected using contrast correction or boosting techniques. We plan to investigate the use of these techniques with high Gamma values to see if higher amounts of energy can be saved while preserving quality levels.

6. RELATED WORK

Many handheld-device-based power saving techniques have been reported in the literature. They attempt to reduce the power consumption of various components of the mobile devices.

In Azevedo et al. [6], a number of architectural and software compiling strategies were proposed to optimise system cache and external memory access. Dantu et al. [14], specifically aims at MPEG-based applications. It proposes to scale the processor voltage and frequency to provide the necessary computing capability for decoding each video frame. In Chandra et al. [10], Stemm et al. [33] and Anand et al. [1], the power consumption of network interfaces (NICs) are optimised.

Dynamically dimming the backlight is considered an effective method to save energy consumed by the mobile device displays [30, 11, 13]. The resultant reduced fidelity can be compensated for by scaling up the pixel luminance. The luminance scaling, however, tends to saturate the bright part of the picture, thereby affecting the fidelity of the video quality. Doing pixel-by-pixel transformation for every frame for luminance scaling is computationally expensive for mobile devices and could result in less energy savings (due to high CPU costs) and lower frame rates which makes it impractical for games as they demand high frame rate.

In Chang et al. [11], a dynamic backlight luminance scaling (DLS) scheme is proposed. Based on different scenarios, three compensation strategies are discussed, that is, brightness compensation, image enhancement, and context processing. However, their calculation of the distortion does not consider that the clipped pixel values do not contribute equally to the quality distortion. In Cheng et al. [13], a similar method, namely, 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.

In Pasricha et al. [30], the authors introduced the concept of a group of scenes (GOS) which defines the granularity at which backlight compensation is performed. GOS is defied 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. The video frames are convolved with a high pass filter to minimise impact on picture detail (loss of contrast) after aggressive luminosity compensation. Cheng et al. [12], employs a technique to incorporate video quality into the backlight switching strategy and proposes a quality adaptive backlight scaling (QABS) scheme. The backlight dimming affects the brightness of the video. Therefore, QABS only consider the luminance compensation such that the lost brightness can be restored.

A more efficient way of backlight scaling using tone mapping is proposed by Iranli et al. [23]. Their method is amenable to highly efficient hardware realisation because it does not need information about the histogram of the displayed image. In addition, many researchers have worked on automatically using tone mapping to compress the high dynamic range of images to a lower dynamic range [16, 17, 26, 19]. Finally, the benefits of different tone mapping operators have been compared using both subjective evaluations [8, 9] and SSIM based objective evaluations [37].

Our solution differs from all the previous works. We are focusing on game applications which have high real-time constraints. Hence, computationally intensive previous solutions are not practical. We exploited the computationally efficient Gamma function, which exists in most modern 3D games, to save display energy. Instead of generating new tone mapping operators as described in Iranli et. al [23], we take a different approach which uses an existing operator (Gamma function) with different parameters.

Our work also differs as we evaluated our system using a careful user study, as user perceived quality is usually more important than any kind of objective measurements. Our study also provides acceptable threshold levels for trading-off quality for power savings. Previous brightness compensation techniques linearly increase the brightness of the image to compensate for the LCD dimming. However, linear techniques cannot save large amounts of energy as they start suffering from clipping or saturation [12] effects which distorts the image heavily. Our use of Gamma's non-linearity slows down saturation heavily and hence, it allows more energy savings with minimal quality loss.

7. CONCLUSION

We have presented our dynamic adaptation technique that saves display energy while playing fast-paced real-time 3D games. Our solution exploits the Gamma function provided by those games to dim the LCD backlight and save display power. We described how we built the complete system with careful user input to ensure that user perceived quality is not degraded due to the dimming of the LCD backlight. We also presented objective measures to support our claims at every step. Finally, we showed, via actual system-level energy consumption measurements on laptops and smartphones coupled with a user study involving 60 participants, that we can save up to 68% of the LCD display's power consumption with minimal quality loss. Our future work includes improving our results further by applying contrast enhancement techniques, and designing a dynamic power management scheme for OLED displays by exploiting the unique characteristics of OLED technology.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] Anand, B., Ananda, A. L., Chan, M. C., Long, L. T., and Balan, R. K. Game action based power management for multiplayer online games. *Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld)*, Barcelona, Spain, Aug. 2009.

[2] Anand, B., Sebastian, J., Ming, S. Y., Ananda, A. L., Chan, M. C., and Balan, R. K. Pgtp: Power aware game transport protocol for multi-player mobile games. *Conference on Communications and Signal Processing (ICCSP)*, Calicut, India, Feb. 2011.

[3] Anand, B., Thirugnnam, K., Long, L. T., Pham, D.-D., Ananda, A. L., Balan, R. K., and Chan, M. C. ARIVU: Power-aware middleware for multiplayer mobile games. *Workshop on Network and Systems Support for Games (NetGames)*, Taipei, Taiwan, Nov. 2010.

[4] Art by Plunkett. *Game Gamma Explained.* http://artbyplunkett.com/Unreal/gamma.html.

[5] Atomic Blue. *PlaneShift - A 3D Fantasy MMORPG.* http://www.planeshift.it/.

[6] Azevedo, A., Cornea, R., Issenin, I., Gupta, R., Dutt, N., Nicolau, A., Veidenbaum, and A. Architectural and compiler strategies for dynamic power management in the copper project. *Conference on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA)*, Maui, Hawaii, Jan. 2001.

[7] Čadík, M. Online comparison of tone mapping operators. http://www.cgg.cvut.cz/members/cadikm/tmo/.

[8] Čadík, M., Wimmer, M., Neumann, L., and Artusi, A. Image attributes and quality for evaluation of tone mapping operators. *Pacific Conference on Computer Graphics and Applications*, pages 35–44, Taipei, Taiwan, Oct. 2006.

[9] Čadík, M., Wimmer, M., Neumann, L., and Artusi, A. Evaluation of hdr tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32:330–349, June 2008.

[10] Chandra, S. Wireless network interface energy consumption: implications for popular streaming formats. *Multimedia Systems*, 9:185–201, Aug. 2003.

[11] Chang, N., Choi, I., and Shim, H. Dls: dynamic backlight luminance scaling of liquid crystal display. *Transactions on Very Large Scale Integration Systems*, 12(8):837–846, Aug. 2004.

[12] Cheng, L., Mohapatra, S., El Zarki, M., Dutt, N., and Venkatasubramanian, N. Quality-based backlight optimization for video playback on handheld devices. *Advanced MultiMedia*, 2007:4–4, January 2007.

[13] Cheng, W.-C., Hou, Y., and Pedram, M. Power minimization in a backlit tft-lcd display by concurrent brightness and contrast scaling. *Conference on Design, Automation and Test in Europe (DATE)*, Paris, France, Feb. 2004.

[14] Choi, K., Dantu, K., Cheng, W.-C., and Pedram, M. Frame-based dynamic voltage and frequency scaling for a mpeg decoder. *Conference on Computer Aided Design (ICCAD)*, San Jose, California, Nov. 2002.

[15] Dong, M., Choi, Y.-S. K., and Zhong, L. Power modeling of graphical user interfaces on oled displays. *Design Automation Conference (DAC)*, San Francisco, California, July 2009.

[16] Drago, F., Myszkowski, K., Annen, T., and Chiba, N. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum*, 22(3):419–426, Sept. 2003.

[17] Durand, F. and Dorsey, J. Fast bilateral filtering for the display of high-dynamic-range images. *Transactions on Graphics*, 21:257–266, July 2002.

[18] Eladawy. *Contrast Error Distribution Measurement for Full Reference Image Quality Assessment.* www.eladawy.com/resources/Others/Doc179.pdf/, Jul. 2009.

[19] Fattal, Lischinski, and Werman. Gradient domain high dynamic range compression. *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, San Antonio, Texas, July 2003.

[20] Gu, Y. and Chakraborty, S. A Hybrid DVS Scheme for Interactive 3D Games. *Real-Time and Embedded Technology and Applications Symposium*, St. Louis, Missouri, Apr. 2008.

[21] Gu, Y. and Chakraborty, S. Power Management of Interactive 3D Games Using Frame Structures. *Conference on VLSI Design*, Hyderabad, India, Jan. 2008.

[22] Id Software. *Quake 3 Arena Source Code.* http://ioquake3.org/, Jul. 2010. (Version 3.21).

[23] Iranli, A. and Pedram, M. Dtm: dynamic tone mapping for backlight scaling. *Design Automation Conference (DAC)*, Anaheim, California, June 2005.

[24] kwaak3. *Port of Quake3 to Android.* http://code.google.com/p/kwaak3/.

[25] Li, X. and Cai, J. Robust transmission of jpeg2000 encoded images over packet loss channels. *Conference on Multimedia and Expo (ICME)*, Beijing, China, July 2007.

[26] Mantiuk, R., Myszkowski, K., and Seidel, H.-P. A perceptual framework for contrast processing of high dynamic range images. *Transactions on Applied Perception*, 3:286–308, July 2006.

[27] Meyer, J. The future of digital imaging - high dynamic range photography. http://www.cybergrain.com/tech/hdr/.

[28] National University of Singapore - Singapore Management University. *Demonstration of Adaptive Display Power Management in Mobile Devices.* http://mmog-pc.ddns.comp.nus.edu.sg/demo/.

[29] NVIDIA Corporation. *Nvidia Tegra - The Mobile Super Chip.* http://www.nvidia.com/object/tegra.html.

[30] Pasricha, S., Luthra, M., Mohapatra, S., Dutt, N., and Venkatasubramanian, N. Dynamic backlight adaptation for low-power handheld devices. *Design & Test of Computers*, 21(5):398–405, Sept.-Oct. 2004.

[31] phidgets Inc. *1127 - Precision Light Sensor.* http://www.phidgets.com/documentation/Phidgets/1127.pdf.

[32] Schonfeld, E. *When It Comes to iPhone Games, What Sells Is Action, Adventure, and Arcade.* http://techcrunch.com/2010/02/25/iphone-games-what-sells-distimo/, Feb. 2010.

[33] Stemm, M., Gauthier, P., Harada, D., and Katz, Y. H. Reducing power consumption of network interfaces in hand-held devices. *Workshop on Mobile Multimedia Communications*, Princeton, New Jersey, Sept. 1996.

[34] Thomos, N., Boulgouris, N., and Strintzis, M. Optimized transmission of jpeg2000 streams over wireless channels. *Transactions on Image Processing*, 15(1):54–67, Jan. 2006.

[35] University of Michigan. *A Power Monitor for Android-Based Mobile Platforms.* http://powertutor.org/.

[36] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *Transactions on Image Processing*, 13(4):600–612, Apr. 2004.

[37] Yeganeh, H. and Wang, Z. Objective assessment of tone mapping algorithms. *Conference on Image Processing (ICIP)*, Hong Kong, China, Sept. 2010.