

CAPED: Context-aware Personalized Display Brightness for Mobile Devices

Matthew Schuchhardt

Northwestern University
shook@shookit.com

Susmit Jha

Intel Corporation
susmit.jha@intel.com

Raid Ayoub

Intel Corporation
raid.ayoub@intel.com

Michael Kishinevsky

Intel Corporation
michael.kishinevsky@intel.com

Gokhan Memik

Northwestern University
gokhan.memik@gmail.com

Abstract

The display remains the primary user interface on many computing devices, ranging from traditional devices such as desktops and laptops, to the more pervasive devices such as smartphones and smartwatches. Thus, the overall user experience with these computing devices is greatly determined by the display subsystem. Ideal display brightness is critical to good user experience, but actually predicting the ideal brightness level which would most satisfy the user is a challenge. Finding the right screen brightness is even more challenging on mobile devices (which is the focus of this work), as the screen tends to be one of the most power consuming components. Currently, the control of display brightness is usually done through a simplistic, static one-size-fits-all model which chooses a fixed brightness level for a given ambient light condition.

Our user study and survey of research literature on vision and perception establish that the simplistic model currently used for display brightness control is not sufficient. The ideal display brightness level varies from one user to another. Furthermore, in addition to ambient light, we identify additional contextual data that also affect the ideal brightness. We propose a new system, Context-Aware Personalized Display (CAPED), that uses online learning to control the display brightness, and is theoretically and practically shown to improve prediction accuracy over time. CAPED enables personalization of brightness control as well as exploitation of richer contextual data to better predict the right display brightness. Our user study shows that CAPED improves the state-of-the-art brightness control techniques with a 41.9% improvement in mean absolute prediction accuracy. Our user study also shows that on average the users had 0.8 point higher satisfaction on a 5-point scale. In other words, CAPED improves the average satisfaction by 23.5% compared to the default scheme.

1. Introduction

The display is the primary user interface in many devices across the computing spectrum. It is used as the primary output interface on

traditional devices such as desktops and laptops. With the advent of the pervasive and ubiquitous computing era, devices such as smartphones, tablets, and smartwatches also use the display as the primary source of input to efficiently utilize their small size and form factor. This trend toward display-centric operation began with resistive and stylus-based PDAs and phones (such as the PalmPilot®), and reached a new level of popularity with modern smartphones and tablets. Thus, in many modern computing devices, the display quality is an important factor in determining the overall user experience.

The relative quality of a mobile display depends on many parameters such as its size, resolution, and brightness. Larger and higher-resolution screens have received widespread attention [1]. Our focus is on controlling the display brightness to improve user experience. Unlike other display characteristics, brightness needs to be dynamically adjusted to suit the user in a given environment. For example, when checking emails in a dark room at night on their phone, a given user would tend to prefer a lower brightness, but when using their phone outside on a sunny day, that same user would tend to prefer a higher brightness. The choice of ideal brightness clearly depends on the ambient light in the surroundings. This led to the inclusion of ambient light sensors on many mobile devices. On most modern smartphones, the brightness of the display is set via the ambient light reported by these sensors [16].

However, this approach has two primary limitations. First, it ignores the difference in brightness preference from one user to another. Not all users are identical and one user might require a brighter screen for the same environment than another user. Second, the current approach does not include external context, apart from ambient light. We define *context* as any data which can be used to better explain a user's current state and their surrounding environment. We did a survey of existing research literature on perception [2, 9, 10] to discover existing knowledge on variance of brightness perception from one individual to another. We also conducted user studies to investigate the need for personalization as well as the need for using additional context information. Our studies presented in Section 2 clearly show that the one-size-fits-all simplistic approach to controlling brightness of displays needs to be improved.

We propose a novel Context-Aware Personalized Display (CAPED) management system. The new system improves the existing state-of-the-art in two ways:

- We discover relevant context apart from ambient light which influences the preferred display brightness level, and include these in our new system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ESWEEK '14, October 12–17, 2014, New Delhi, India.

Copyright © 2014 ACM Copyright 2014 ACM 978-1-4503-3050-3/14/10...\$15.00.
<http://dx.doi.org/10.1145/2656106.2656116>

- We develop an online learning-based approach which enables personalization to individual user preferences. Our algorithm is shown to theoretically and practically improve in accuracy over time as it adapts to user preferences. Further, we show that on average, we would never be worse than the current state-of-the-art technique which uses a fixed mapping for all users.

We validate our new system through a user study conducted on 10 users who were asked to use CAPED-enabled smartphones as their primary device for one week. Our user study demonstrated that the new system could more accurately predict the users' preferred screen brightness level than the existing state-of-the-art. The average improvement in absolute prediction error in CAPED is 41.9%. Users were also asked to rate the default and CAPED systems after a day of use. On a 5-point scale of satisfaction reported by the users, CAPED improves the satisfaction by 0.8 points on average.

The rest of the paper is organized as follows. We discuss the need for personalization and the need for including additional context information in Section 2. In Section 3, we describe some background on mobile display subsystems. In Section 4, we describe our proposed system and its different components. We explain the experimental setup used in our user studies in Section 5. We present our results in Section 6, introduce some related work in Section 7, and give a brief conclusion in Section 8.

2. Motivation

In this section, we present our motivation for the creation of an improved adaptive brightness model. The two primary pieces of motivation that we provide are the need for personalization and the need for the inclusion of additional contextual information in display brightness models.

2.1 Importance of Personalization

One of the most significant issues with current display brightness models is that they are generalized to the average user, and allow little room for personalized, per-user settings. These models assume that all users have the same screen brightness requirements, and so screen brightness is controlled accordingly. We don't believe that this is a sufficient approach to display brightness control, and we present some supporting evidence here.

One visual perception metric for mobile displays is that of *readability*. There are many ways to define a display's readability, but here we consider the metric *RVP* (relative visual performance), described by the standards organization CIE [2], and further explored in the work by Kelley et al. [10]. *RVP* measures the accuracy and speed with which a user can read text at a specific detail size, ambient light level, and contrast ratio. Some *RVP* curves for discerning detail at a 1.5' (minutes of arc) are presented in Figure 1.

One notable feature of the graphs in Figure 1 is that they confirm that *RVP* is highly dependent on the contrast ratio C . It's also interesting to note that readability increases with the average luminance, *but only when the contrast ratio is held constant*. Since increased ambient luminance decreases the contrast ratio on LCDs, increased ambient luminance will actually reduce *RVP*.

The need for personalization can be seen by comparing the *RVP* curves for these three graphs. These graphs contain aggregated data for subjects of varying ages. As these graphs show, the required contrast for a given *RVP* level increases significantly as users age. This point alone suggests that a generalized model can be improved upon with a more personalized model; age is not considered at all in generalized brightness models. Furthermore, it is important to note that these graphs contain data aggregated across a number of users for a given age. Simply knowing one particular user's age does not mean that that user's individual *RVP* curve can accurately be

determined; there is a large amount of additional variability even between users of a similar group.

For example, visual acuity can vary from user to user. Visual acuity is a measurement of the clarity of a person's visual system, and is dependent on a variety of factors, including the quality of the focused image on the retina, the proper functioning of the retina itself, and the ability of the nervous system and brain to transmit and interpret the visual data [5]. A user with poor visual acuity will have worse *RVP* measurements than a user with good visual acuity in the same ambient light and contrast range.

There is even variability among two users with similar visual acuity. Contrast sensitivity is a measurement of how well a user can discern contrast in a given scene [9]. Even a user with perfect visual acuity may struggle with contrast discernment tasks. Since contrast sensitivity is directly related to the contrast curves in the *RVP* metric, this further complicates the feasibility of a generalized brightness model.

To further experimentally motivate our work, we run an initial, controlled user study. To do this, we brought a series of users into our lab. The users were seated with a Google[®] Nexus[™] 4 smartphone. Users were then asked to indicate their preferred brightness levels while we artificially manipulated the surrounding ambient light levels and on-screen images.

The results from this study are shown in Figure 2. This data contains 4 dimensions (the user, the displayed image, the ambient light level, and the user's brightness preference). To show the underlying trends in this data, in Figure 2a we average the results across different images, while in Figure 2b, we average our data across users.

As the Figure 2a shows, not only do users' display preferences differ from one another significantly, but they also differ from the default brightness model on the device. There is a general positive correlation between brightness and ambient light, but some of the users do exhibit either negative or little correlation of their preferred brightnesses with ambient light.

Although *RVP* is solely based on textual readability, our motivating study looks at brightness requirements for both images and textual data. Figure 2b suggests that required screen brightness varies not just for text, but also for images. It's interesting to note that the actual on-screen image impacts the preferred brightnesses; some on-screen content demands higher or lower brightness levels than others for certain users.

Given the fantastic complexity of any individual user, we believe that a generalized model of user brightness preferences is infeasible at best, and impossible at worst. Because of this, we propose that to accurately predict screen brightness levels, we must use a personalized, per-user brightness model. Furthermore, we also believe that there are external factors besides the visual system and readability which may significantly impact an individual's brightness requirements at any given point in time. We outline these external factors in Section 2.2.

2.2 Contextual Data Inclusion

In Section 1, we defined context as any data which can be used to explain a user's current state and their environment. Currently, baseline models for screen brightness requirements are solely based off of contextual data regarding ambient light. Ambient light contextual data allows the model to attempt to provide a more constant level of readability to the user. However, we believe that there are other contexts that are important in accurately predicting preferred screen brightness. Here, we outline the contextual data that we include in our proposed system, and provide some intuition behind the decision of using the context in our model. This is not meant to be an exhaustive list of all possible contextual data, but as a reasonable starting point for a more contextually-aware system.

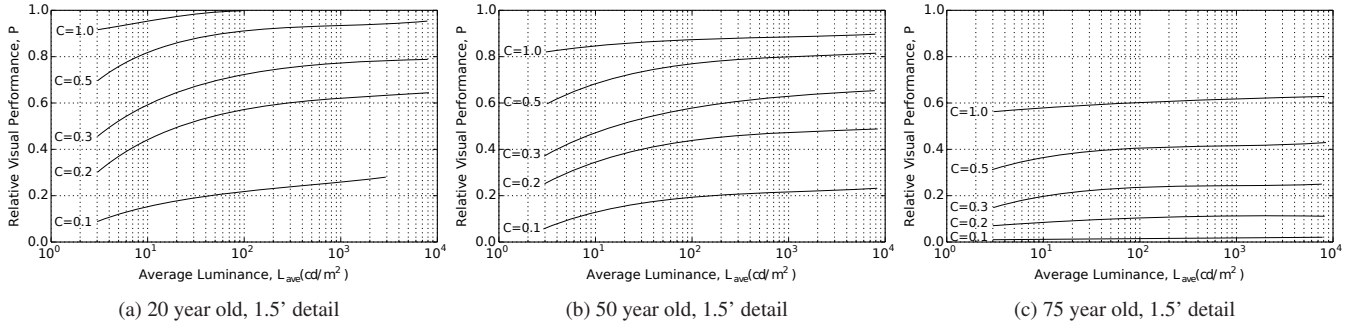


Figure 1: Aggregated readability curves [10] for users of varying ages, generated using data from CIE [2]. As users age, their ability to discern detail drops significantly at a given luminance and contrast ratio level. Contrast ratio and luminance are both strong predictors of readability.

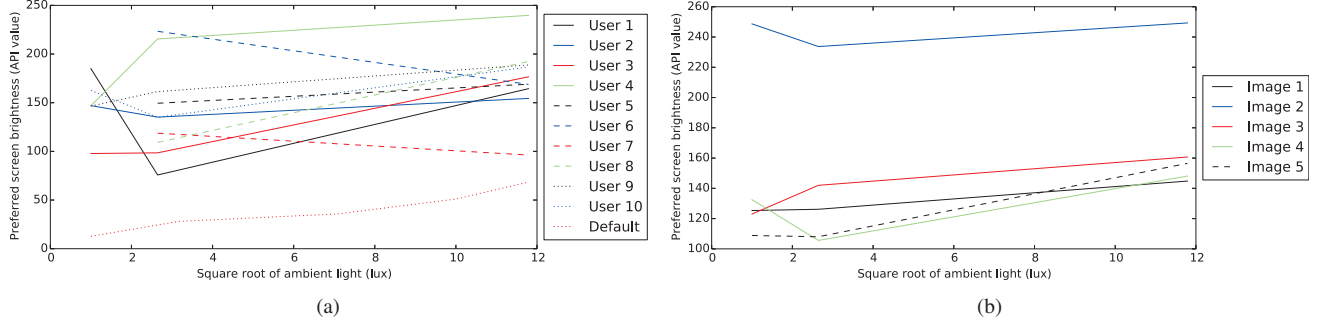


Figure 2: Results of initial study which analyzed users' brightness preferences for a selection of on-screen images and ambient light levels. Figure 2a suggests that user brightness preferences differ from one another, and from the manufacturer's default brightness model. Figure 2b suggests that image content significantly impacts brightness preferences.

Circadian rhythm is a 24-hour cycle that many biological processes are based around. Excess artificial light is known to have the potential to disrupt the regularity of this cycle [6]. This can make it difficult to fall asleep, stay asleep, wake up in the morning, or function at full energy throughout the day. Because of this, we suspect that users may have differing brightness requirements depending on where the sun is in the sky. We don't believe that this context should be used in a generalized model, because people have their own natural tendency toward being more active at night or during the day (night owls vs early birds) [12].

We also believe that the duration a display is active can be a good predictor of brightness requirements. We propose using device on-screen time, which is a measure of how long it has been since the user turned their screen on, as a useful piece of contextual data. Since phones provide their own light source, the human eye adjusts to this source of light altering its physiological characteristics [3]. We intuitively expect that as users become acclimated to the phone's brightness, they may have varying brightness requirements.

Accelerometer data is another potentially useful piece of contextual data. The more that a phone (and the user) moves around, the harder it becomes to see the screen, reducing RVP at a given screen brightness level [15].

A similar context, activity characterization, may also improve brightness predictions. Activity characterization uses a combination of on-device sensors to predict what a user is currently doing (e.g., still, walking, biking, driving, or tilting the screen). Because visual acuity or the amount of constant attention that can be given to the display may differ between these activities, we believe this context to be potentially useful.

Battery level is another interesting piece of contextual data. We anecdotally observed that some users adjust their screen brightness depending on how much battery life is remaining. If the remaining battery level is low, the tendency is to dim the screen to extend the remaining on-screen time, which suggests that this context may have some predictive power.

Finally, we include location data as an input context in our model. We believe that the user's current location (work vs home, etc.) may impact a user's display brightness requirements.

In Section 2.1, we noted that on-screen content can be a strong predictor of required display brightness. However, for the reasons specified in Section 4, CAPED is implemented as an application-level piece of software rather than integrated at the platform level. Because of the UI lag that including display content as a context causes, we don't include display content contextual data for this study. We may, however, explore the inclusion of screen content at the platform level in the future.

3. Background

In this section, we introduce some common mobile display technologies, and explore their optical characteristics and their relationship with ambient light.

3.1 Modern Display Technologies

There are many mobile display technologies, and we outline some of the more prominent ones in this section. The remainder of the paper will solely focus on LCDs, but OLED displays behave in an optically similar manner, suggesting that this work can also be applied to OLED displays. Because reflective displays have no light source of their own, they are only included for discussion.

3.1.1 LCD Displays

LCDs are the most widely used and mature of the mobile display technologies. The two most important active elements of an LCD are the backlight and the liquid crystal matrix. The backlight is just as it sounds – it is simply a light source which rests at the back of the LCD. On top of the backlight lies the liquid crystal matrix. The liquid crystal matrix controls the colors on the screen.

The liquid crystal matrix contains an array of red, green, or blue liquid crystal light filters. These filters are controlled by a connected voltage line, which alters the crystalline state of each point. Depending on the crystalline state of the liquid crystal material, each colored filter appears either transparent (appearing red, green, or blue), opaque (appearing black), or somewhere in-between. An individual pixel is made up of the combination of three (or more) of these red, green, and blue points, and depending on how the colored elements are combined, the whole gamut of the color spectrum can be produced. A modern display contains millions of these individual pixels.

3.1.2 OLED Displays

OLEDs have similar optical characteristics to LCDs, but use vastly different technology. Instead of using a backlight and a configurable color filter, a pixel on an OLED is comprised of a set of extremely small LEDs, each of which emits its own red, green, or blue light. By adjusting the brightness of each of the red, green, and blue LEDs, the visible color spectrum can be produced.

LCDs have slightly different operating characteristics than OLED displays. First of all, LCD pixels can never be completely black; some of the backlight always leaks through the pixel, even when the crystalline state is as opaque as possible. OLEDs *can* be completely turned off, and so have better black level characteristics. Secondly, LCD power consumption is primarily dependent on the backlight, while OLED power is dependent on the state of each individual pixel. Because of this, the color content of the on-screen image of an OLED impacts the power consumption, while an LCD at a given backlight level always uses the same amount of power.

3.1.3 Reflective Displays

Reflective displays, commonly used in eBook readers, do not provide their own light source. Instead of pixels, a reflective display contains a matrix of individual points of electrically controlled pigment, which is optically similar to a piece of paper. Since the pigments absorb an approximately constant percentage of incoming light, a good display contrast is maintained even in direct sunlight. However, reflective displays have slow refresh rates, and perform poorly with dynamic screen content.

3.2 Display Subsystem

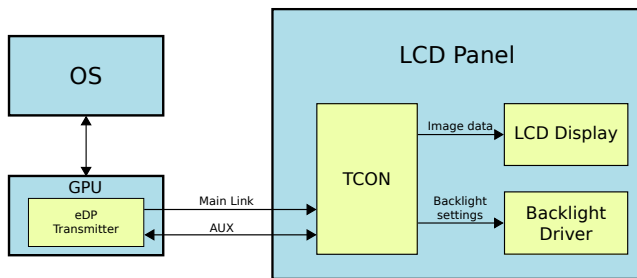


Figure 3: Diagram for the Embedded DisplayPort™ display subsystem [21]. This organization of subsystem blocks is typical for modern mobile devices.

A mobile LCD consists of a number of logical blocks interacting with one another. We give a high-level overview of a typical display subsystem in Figure 3.

As the figure shows, the primary blocks include the operating system, the GPU, the LCD’s *TCON* (Timing CONtroller) board, and the LCD panel itself. The process of displaying an image on the screen begins with the OS and GPU. Using data provided by the operating system, the GPU is tasked with rendering what will eventually become an image on the mobile display. Once that image is rendered, it is passed along from the eDP (embedded DisplayPort) transmitter on the GPU to the TCON board via the main link, which handles the synchronization and populating of data on the display panel.

One important mobile optimization is the self-refresh system [21]. Panel self-refresh takes advantage of the fact that oftentimes, the image on a mobile screen isn’t changing. For instance, if a user is reading text, viewing a static image, or waiting to interact with the UI between animations, the display image is completely static. Instead of the GPU constantly providing the display panel with updated image information (again, even though the image is static), the panel self-refresh system allows the GPU to enter a low-power state until a new image actually needs to be rendered. Meanwhile, the TCON board internally handles displaying the same constant image on the display without requiring input from the GPU, which saves both GPU and bus transfer power.

A second important mobile display optimization is automatic backlight brightness control. It is common for mobile displays to allow programmatic setting of the backlight brightness via the operating system. This enables the operating system to control the backlight brightness, which allows screen brightness optimization in a wide variety of ambient light ranges. As of the Embedded DisplayPort™ v1.2 specification, this data is passed via the eDP transmitter’s AUX data line to the TCON panel, which is in turn passed to the backlight driver. This backlight control is the element of the display subsystem that we focus on improving in this work.

3.3 Display Optical Characteristics

We mentioned in Section 3.2 that a programmatically controlled backlight allows for improved viewing characteristics in a wide range of ambient light ranges. But why does ambient light have such an impact on the legibility of a display?

Contrast ratio, as we explained in Section 2.1, is a significant indicator of the readability of a display. There are a variety of ways to express contrast ratio, but a generalized equation for the contrast ratio of a display is given by Equation (1) [10], where L_{max} is the highest achievable luminance of the display, and L_{min} is the lowest achievable luminance of the display. *Luminance* is defined as the amount of light leaving a given surface, while *illuminance* is defined as the amount of light striking a surface from some light source. Luminance is dependent on the surface’s reflective characteristics.

$$C = \frac{L_{max} - L_{min}}{L_{max}} \quad (1)$$

Thus, as the difference between the brightest and darkest parts of the screen increases, so does the contrast ratio.

However, if contrast ratio is determined by the darkest and brightest parts of the screen, then what does ambient light have to do with this equation? It’s important to note, that even though contrast ratio is dependent on how bright and dark the screen can get, the surrounding ambient light impacts these light and darkness values. This is shown graphically in Figure 4.

When ambient illuminance E comes into contact with an LCD, it doesn’t just dissipate; some portion of that light is reflected back to the viewer. The amount of reflected light is scaled by the display’s reflective coefficient ρ . This reflected light is seen together with

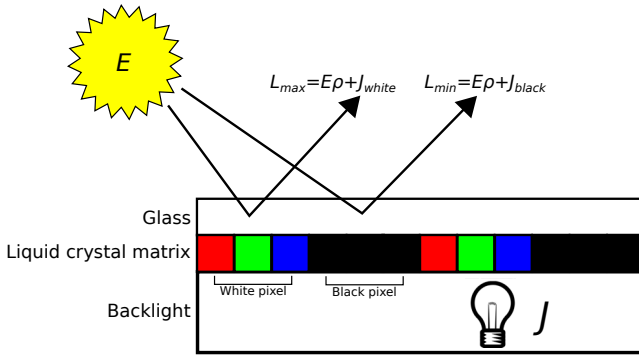


Figure 4: Representation of light-display interaction. Ambient light directly impacts LCD contrast ratio.

the device’s emitted light J , which impacts the display’s contrast ratio. To illustrate this point: in a perfectly dark room, C is solely a function of just the screen’s white and black characteristics, and is at its maximum possible value. In a brightly lit area, however, the ambient light increases the luminance of the screen’s white and dark regions, which can significantly reduce the contrast ratio.

4. System Description

As outlined in the previous section, there are a number of shortcomings that current adaptive screen brightness systems currently have. The primary deficiencies we identified are that current adaptive screen brightness systems are one-size-fits-all, and that providing additional contextual data can improve adaptive screen brightness prediction accuracy. In this section, we describe CAPED, our proposed system for addressing these deficiencies.

4.1 Proposed Model Description

To describe our model for predicting user satisfaction, we first begin with the manufacturer default ambient brightness model, and gradually modify that model to incorporate personalization and context awareness in the model. As shown visually in Figure 5a, the sole input to the default prediction model is ambient light information. A simple, fixed function is then applied to the ambient light values, and a predicted value for that ambient light level is generated.

This model is completely static, which means that no personalization can be performed. To address this, we allow users to directly provide our model with preferred brightness levels. With each new brightness preference indication, the adaptive model is updated to better match the user preference.

It’s important to note that one of our top priorities in this system was to improve the adaptive brightness model, but only if we can do so without negatively impacting the user experience. Thus, we do not want to interrupt users or disturb their device session in any way to get brightness preference information. Instead, users make their brightness indications only when they are dissatisfied with the brightness state. Their indications are easily made via a button in the notification area, which displays a slider that allows them to select a preferred brightness level.

The second aspect of the default model we aim to fix with CAPED is that the only context used as an input to the model is ambient light. In Section 2.2, we presented evidence that ambient light is not the only important predictor of preferred screen brightness. To allow additional contexts as inputs to our model, CAPED enables an arbitrary number of contexts as input features, targeting screen brightness as our predicted output. This proposed system is described in Figure 5b.

Thus far, we have been intentionally vague about this “adaptive model” which is learning user brightness preferences with some number of contexts as input. There are any number of possible models which can be used to predict an output given some set of inputs. Something as simple as a linear regression may be able to accurately predict an output, or it may require a more advanced general-purpose model such as support vector machines, or even a non-linear model like decision trees. Furthermore, an adaptive model doesn’t even need to be a general-purpose model; if some amount of domain knowledge is available about the output, a hand-crafted model may also be effective. Because we don’t know exactly what model will best suit a given user, we don’t want to pre-select a single adaptive model in our system. Instead, we employ the use of several simultaneous learning models, each of which provides their own outputs. This allows us to eventually prefer predictions from the most accurate sub-models over those from the least accurate ones. We more clearly define this system in the following section.

4.2 Online Model Composition

Given a set of learning models for predicting user brightness preferences from contextual data, we need a meta-algorithm to do online selection of models to minimize prediction errors. Some models may be better predictors for certain users and in some context. Automatically inferring which model to use for prediction is a classical online learning problem. Blum provides a detailed survey on online learning, describing different techniques to select prediction models adaptively [4]. One of the classes of online algorithms is the weighted majority algorithm [14]. This algorithm is a binary classification algorithm which contains a pool of individual binary classifiers, each of which classifies data independently. In addition to these independent classifications, each sub-algorithm contains a weight. As training data is introduced to the sub-algorithms, these weights are either increased or decreased in value by some function depending on if the classification was made correctly or not. To classify a new piece of data, the weighted majority algorithm calculates the weighted value of the “0” predictions to that of the “1” predictions from each of the sub-algorithms. The prediction with the higher weight is the selected prediction from the model. This algorithm is useful to CAPED because the weighted majority algorithm gives upper bounds on the number of incorrect predictions that the meta-algorithm contains.

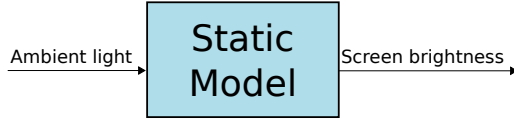
However, the weighted majority algorithm [14] only predicts binary data; screen brightness is continuous data, and so we need to modify the weighted majority algorithm to work with a non-binary output. We use the continuous variant of the weighted majority algorithm proposed by Vovk [20]. Instead of using the weighted summation of each sub-model’s predictions, each prediction is instead a weighted combination of the sub-models, where \hat{p} is an individual prediction, N is the number of experts, w is a weight, and f is an individual sub-model’s prediction, as shown in Equation (2). The output from this formula is continuous rather than binary.

$$\hat{p}_t = \frac{\sum_{i=1}^N w_{i,t-1} f_{i,t}}{\sum_{i=1}^N w_{i,t}} \quad (2)$$

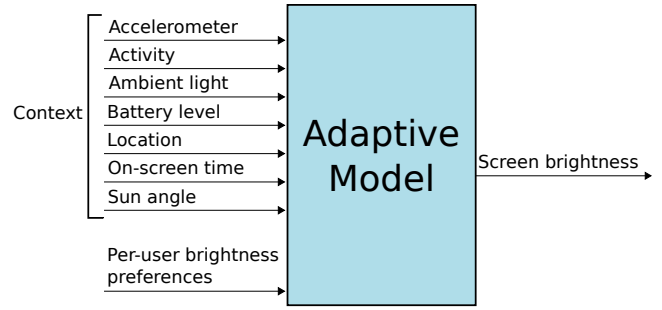
As new training data is added to the online meta-algorithm, the weights on each sub-model are updated depending on how close their prediction was to the result, where y is the actual outcome and l is some loss function:

$$w_{i,t} = w_{i,t-1} e^{-\eta l(f_{i,t}, y_t)} \quad (3)$$

The most important piece of this algorithm for CAPED is the upper bounds on accuracy that it provides. Because we don’t know in advance which algorithms will most accurately predict user state, we wish to allow this system to gravitate to the most accurate classifiers.



(a) Default model: The only input to this model is ambient light, and the model is static.



(b) CAPED: we add additional inputs to the system, and also allow users to provide their preferred brightness levels at a given context. CAPED is updated as additional input is provided.

Figure 5: Baseline and proposed adaptive brightness management.

Over time, the number of errors in the meta-algorithm is known to converge to the number of errors in the best sub-algorithm. See [20] for theoretical proof of convergence. Thus, given a sufficiently large number of predictions, the meta-algorithm will not do worse than any of the sub-models. This is represented by Equation (4), where \hat{L} represents the error from the online model composition, and J is the pool of experts.

$$\hat{L}_n \leq \min_{i \in J} L_{i,n} + \sqrt{\frac{\ln(N)}{t}} \quad (4)$$

In our specific implementation, since we include the manufacturer’s default brightness model as one of the sub-models, this implies that on an average after adequate training points, we will not perform worse than the default brightness model. This is important in situations where none of the other sub-models are able to accurately capture the user’s preferences, or if the user gives noisy inputs to the system.

Another interesting side-effect of this meta-algorithm is that the most heavily weighted sub-algorithms weights can change throughout the system’s lifetime. Because a heterogeneous set of sub-models are used in this system, it may happen that some algorithms work sufficiently well with a small training set, while others work better with a large training set. This allows the meta-algorithm to prefer sub-models which have the best prediction accuracy, even if the most accurate models change over time.

4.3 CAPED System Architecture

In this section, we describe the specifics on how CAPED is integrated with the Android Nexus 4’s operating system. A summary of this architecture is shown in Figure 6.

CAPED is programmed entirely in Java, and exists as a user application which manifests itself as a conglomeration of a number of background services and user-facing activities. CAPED’s contextual data gathering is accomplished via standard Android userspace API calls. We described the motivation for including the various contextual data in Section 2.2; we now give specifics as to how these contexts are collected, and what data we specifically extract from them. All of the contextual information that we gather are from either on-board sensors, third-party Google APIs, or calculated programmatically. The accelerometer, ambient light, and battery contexts are all sourced from on-device sensors via the standard Android application API. The activity characterization and raw location data is obtained from Google’s location and activity characterization APIs. We perform some additional calculations to cluster the locations into clusters with a maximum radius of 0.5km. Finally, on-screen time and the sun’s angle are calculated using the device’s

clock. Each of these contexts are calculated in their own individual thread, which allows us to configure update rates individually.

CAPED’s meta-algorithm is implemented as follows. The meta-algorithm contains a pool of sub-models, each which has a weight associated with it. When a prediction is run, then all of the current contextual data is packaged into a data structure, passed to all of the sub-models individually, and each sub-model makes a prediction using that contextual data. The predictions from all of the sub-models are gathered, multiplied by their weights, and that result is combined into one final meta-output.

We generate display brightness predictions at a rate of 1Hz. To control the display brightness, we use API calls which manually set the device’s screen state using the meta-algorithm’s meta. The device’s default adaptive brightness system is disabled when CAPED is active.

To add training data to CAPED, a button is displayed in the device’s notification area. Selecting this button displays a dialog box which allows the user to select their preferred brightness level. When the user selects their preferred brightness level, a series of events happen. First of all, that brightness indication is packaged up along with the current contextual data. The sub-models then each run an individual prediction (without using the new training data yet), and depending on how close their prediction was to the actual preferred output, the sub-models’ weights are updated accordingly. Finally, the sub-models themselves are updated with the new training data. None of the general purpose models we use are “updatable” models; this means that with each new piece of training data, the adaptive sub-models are rebuilt from scratch using the cumulative set of training data. Using updatable models could optimize the amount of time that it takes to add a new piece of training data, but wouldn’t have any impact on the model accuracy. Even on larger training sets (≥ 1000 instances), generating a sub-model takes less than a second, and this only occurs when the user makes a new brightness indication, so we don’t pursue updatable models in this work.

The general purpose sub-models are provided by the Java-based Weka [8] machine learning library. Weka is shipped as a Java-based machine learning solution with GUI support; this GUI support had to be stripped out to be compatible with Android. We also include the manufacturer’s default model as one of the sub-models. This sub-model is completely static; it doesn’t change as new training data is added.

We developed CAPED as a user application for the sake of improved portability, rapid development, and ease of deployment (allowing us to eventually distribute the application to the public). However, there are some advantages that could be provided by implementing this system at the platform level. First of all, a

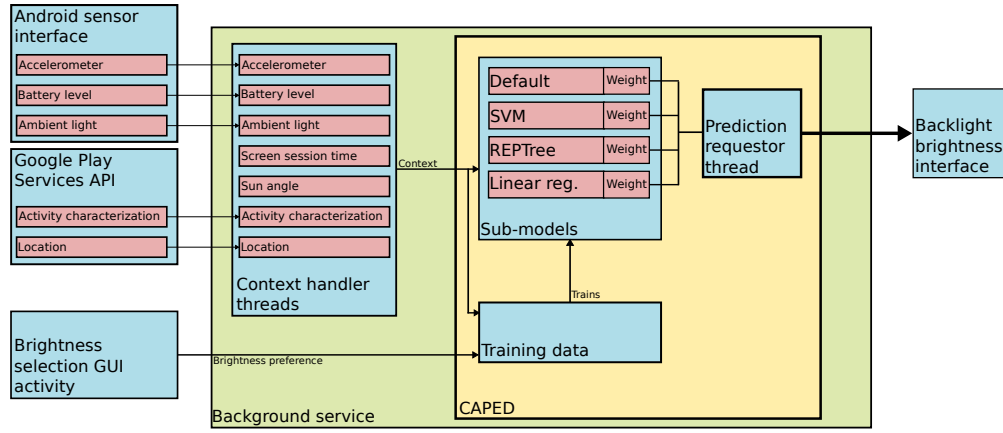


Figure 6: Integration of CAPED with the Android operating system.

platform-level implementation would allow for direct control of the screen brightness. Manually controlling the brightness via API calls means that the operating system ultimately controls the transition between brightness levels. A platform-level implementation of CAPED would allow for fine-grained control over the brightness transitions, but these transitions are out of scope for this work. Secondly, accessing the display’s image content at the application level is a computationally intensive process which introduces severe UI lag into the system when display content is analyzed at the rates that our system requires. Platform-level access could make the inclusion of screen content as a contextual input feasible by reducing the CPU and bus bandwidth overhead of the calculations. Finally, a platform-level solution could be more deeply integrated into the brightness subsystem, which would allow us to hide more of the complexity of the brightness controls and training from the users.

5. Experimental Setup

In this section, we describe the experiments we conduct to measure CAPED’s effectiveness. We perform user studies on a set of 10 smartphone owners. Participants were gathered via fliers advertising the study.

Each of the users was provided with an Android Google Nexus 4 smartphone to use as their primary mobile device for the period of one week. We used a standard device for all of the studies because display brightness models vary between devices, and this allows us to better understand the user preferences compared to the default model. Users’ existing cell phone plan was used to provide data access on the Nexus 4 device. The users were instructed to use this device as if it was their own; they were allowed to install any applications on the phone that they would typically use, and were told to customize any additional settings on the device as they see fit. CAPED is installed on each of the devices.

The number of users in this study was limited to 10 because the study lasts a number of days per user, and because we provided the equipment for the users. We wanted to have all users using the same device model, not because of any technological limitation (CAPED has been successfully run on many other models), but because we wanted to analyze how different users interact with the same display interface. Our sample size of 10, although not large enough to precisely describe every way that a person could possibly interface with CAPED, is certainly large enough to give results that show some variety in users, while repeatedly demonstrating the effectiveness of CAPED.

While they were using the device, the users were instructed to indicate their preferred brightness levels whenever they were dissatisfied with the display’s brightness. Each time the user made a brightness selection, that preference, along with the current contextual data, is added to CAPED’s training set.

CAPED uses four sub-models: the default Nexus 4 brightness model, an SVM regression model, a linear regression model, and a decision tree model. The default model is initially given a weight of “1”, while the adaptive models are given weights of “0.0001”. We weight the models this way because the default model has been pre-calibrated to suit the average user, and we expect that it will have the best accuracy without any training data provided. If the generic machine learning models begin to predict brightnesses more accurately than the default model, they will become more highly weighted than the default model, and those predictions will contribute more to the model’s output.

To better gauge how users perceive our brightness algorithm, we split our study into a few different phases. During the first three days of the study, CAPED is used to control the backlight brightness, and the users’ brightness indications are used to train the model; this is how CAPED would function in a real system. On the final four days, two of the days use our brightness prediction model, and two of the days solely use the manufacturer’s default brightness model, in a randomized order. At the end of each of those 4 days, users are asked to rate their satisfaction with the screen brightness during the previous 24 hours. This allows us to subjectively gauge how satisfied users were with CAPED.

To get more information about the user’s surrounding context, we also log contextual data two times per minute. This data collection is not at all tied to the user indications.

6. Results

In this section, we present our results for describing the effectiveness of CAPED, and compare it to existing screen brightness models.

6.1 Prediction Accuracy Analysis

In this section, we describe the accuracy of CAPED, compared to the accuracy of the default model. During the user study, each user was asked to adjust their preferred brightness settings when they were dissatisfied with the current brightness. For each of these preference indications (and before using this data to update CAPED’s prediction model), we collected the predicted brightness value for the given contexts from the default and from CAPED. From these predictions, we calculated the error for both models, for each indication. These errors were then averaged and are presented in Figure 7.

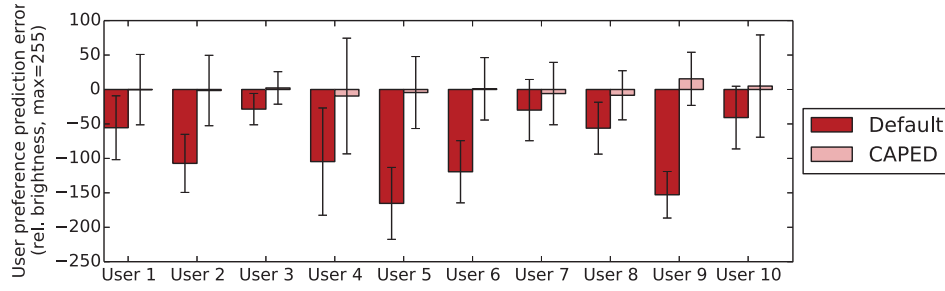


Figure 7: Comparison of prediction model accuracies compared to the actual user-indicated brightness preferences. The mean error is presented in units of relative screen brightness, which is used by the Android brightness settings API. Bars represent one standard deviation. Minimum brightness=0, maximum brightness=255.

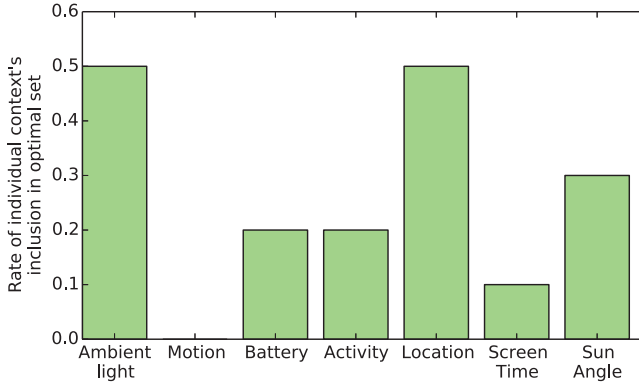


Figure 8: The rate at which each context was included as part of a user's optimal subset of context features. The optimal subset was determined via an exhaustive subset search.

Across all users, the average error of the default model is -86.03, while the average error of CAPED is -0.05. The mean reduction in absolute prediction error, averaged from each user's errors, is 41.9%. Although Figure 7 shows that the default model typically does underpredict the user's brightness requirements, it does not mean that simply increasing the default model's brightness curve would cause the system to perform as accurately as CAPED. It isn't the common case, but there are many instances where CAPED predicts a brightness lower than the default model, while remaining more accurate than the default model.

It is important to also consider the effectiveness of using a meta-algorithm with multiple sub-models, rather than just using a single adaptive model. For a majority of the users, the progression of the weights for each sub-model tends to follow the same general pattern. The default classifier always begins with the highest weight, since we manually initialize the sub-models as such. Since the user requirements typically significantly differ from the default classifier, as the users continue to train the meta-algorithm, one or both of either the SVM regression or the linear regression models quickly becomes more heavily weighted than the default classifier. Furthermore, the decision tree model was overall the most accurate classifier, but only with a relatively high number of indications (at least 15-20 or so). Hence, the decision tree's weight typically remains relatively low until a sufficient amount of training data is provided. This weight trend isn't identical for all users, as some users behaved more linearly than others, but was the most common case.

This progression of the sub-model weights shows the merit of using multiple sub-models, rather than only one. The most accurate

sub-model tends to change over time; the linear models typically perform better with a small amount of training data, while the more complex models tend to perform better with larger data sets. Thus, the best model is actually a combination of the subclassifiers. Furthermore, it is common (in 70% of cases) for multiple classifiers to contribute significant weights to the prediction at the end of the week-long study, rather than completely converging to a single model. This furthers the case for using multiple sub-models.

6.2 Impact of Contextual Data Inclusion

In this section, we describe the impact that the inclusion of additional contextual data has upon the overall prediction accuracy. To accomplish this, we run an offline analysis of the user brightness preference and indication data. Using an exhaustive subset analysis, we determine the subset of contexts for each user which maximizes that user's brightness value prediction accuracy. We then compare the absolute error of running predictions with this optimal subset to the error of running predictions using ambient light as the sole contextual feature. This method allows us to see how much the accuracy can be improved via additional contextual data, as well as determine which contexts have the most predictive power. Using this analysis, we find that using the optimal subset of contexts has a 14.5% lower error rate than using just ambient light as the sole input.

In Figure 8, we present how frequently each given context is included in the various users' optimal subsets. To calculate this, we simply take each user's optimal context subset and count how many times each context is included in the optimal user models. Expectedly, ambient light is one of the most important predictive contexts, but additionally both the current location and the position of the sun are frequently included in the optimal subset model. The motion of the device seems to have a very low level of predictive power.

6.3 User Rating Comparison

For the final four days of the study, two of the days control the display using the default mode, while two of the days use CAPED. Specifically, at the start of each day, we select the model for that day randomly (the users are unaware of this change). The random selection continues until one of the schemes is selected twice, after which point we continue with the other method for the remainder of the 4 days. At the conclusion of each day, the system prompts the user to rate their satisfaction with the system's brightness between 1 and 5, with 5 being the most satisfied. The average ratings for each model and each user are presented in Figure 9.

As the figure shows, CAPED universally either doesn't impact ratings, or improves them. Also interestingly, in general the users with the largest increase in rating tend to be the users with the largest improvement in mean prediction error. This suggests that

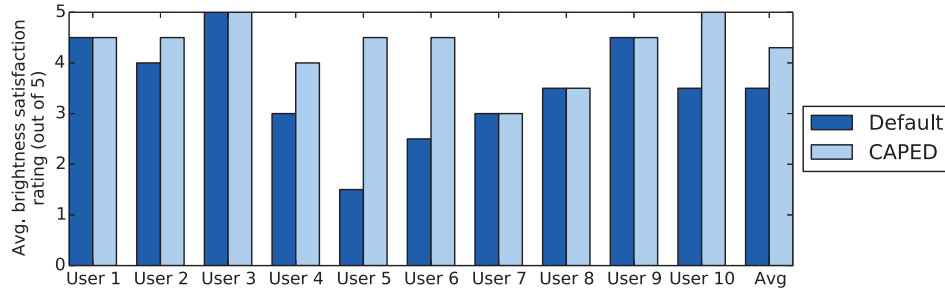


Figure 9: Comparison of user-indicated model satisfaction ratings over a 24-hr period. Users use one of the two models for two days each, for a total of 4 ratings. Mean ratings on a 1-5 scale are presented here.

the prediction error is, indeed, a strong contributor to satisfaction with the screen brightness. On average, CAPED improves the default scheme’s subjective user satisfaction rating from 3.4 to 4.2, or a 0.8-point (23.5%) improvement on a 5-point scale.

6.4 System Overhead

The primary design goal for CAPED was to accurately predict the preferred screen brightness; we intentionally allow CAPED to exceed the default model’s brightness levels if it is closer to the user’s preferences. As Figure 7 suggests, CAPED tends to exceed the default model’s brightness levels. Additionally, as one would expect, the contextual data collection and prediction systems of CAPED have an additional impact on the device’s power consumption. Because increased power consumption is a concern on any battery-constrained device, in this section we examine CAPED’s impact on power.

During the user studies, we periodically logged the device’s contextual data. Even though only one of either CAPED or the default model can be active at a time, we can use the contextual data to “replay” what the screen brightnesses would be over the same periods of time. We additionally generate a linear model which relates the screen’s brightness to the amount of current being discharged from the battery. We create this model by gathering amperage data from the Nexus 4’s internal current sensor (part of the device’s power management IC) at a variety of backlight brightness levels. This model, along with the brightness data of the two schemes, allows us to determine the difference in power consumption due to screen brightness between the two schemes over time. From this data, we then integrate the current flow to find the total difference in mAh (milliamp hours) between the models for each user per day, using their actual usage trends. Finally, because the Nexus 4’s battery has a known capacity of 2100mAh, we then calculate the difference in the percentage of the total battery capacity that is consumed by the screen per day.

CAPED slightly increases the screen’s power consumption for all users. Across our participants, the smallest increase in daily battery capacity usage is 0.5%, while the largest increase in daily battery capacity usage is 5.5%. On average, CAPED will cause the screen to consume an additional 2.1% of the total battery capacity per day due to increased screen brightness. This is a small increase in power compared to the display’s overall power consumption [19]. On average, each user used the Nexus 4 device for 55 minutes each day. Although there is a general correlation between on-screen time duration and increased battery consumption, it interestingly is still heavily dependent on the individual users’ brightness preferences and contextual data. For example, one user had a daily on-screen average of 149 minutes with a total daily battery consumption increase of 3.5%, while another user had a daily on-screen average of 49 minutes, with a 5.5% battery consumption increase.

Another source of power overhead is with regard to the prediction system itself. We compared the system’s idle current draw while the screen was active and set to minimum brightness, with and without CAPED installed. Without CAPED installed, we found the Nexus 4’s average idle current draw to be 212mA. Collecting the same data, but with CAPED running predictions and collecting contextual data, the average current draw is 280mA. However, the majority of this increase is due to the activity characterization collector; with this piece of contextual data disabled, CAPED only draws 228mA. It’s important to note that CAPED only increases power consumption while the display is actually on; the majority of the time, the devices weren’t actively used. Because of this, the amount of total daily battery capacity consumed by CAPED is 2.83% on average with activity characterization, and only 0.67% with activity characterization disabled.

There is also a system overhead that applies to the users themselves; a training system which requires constant interaction but improves the prediction accuracy could still reduce the user’s overall satisfaction with the system. On average, users adjusted the display brightness 4.4 times per day when CAPED was active. This training set size, which is small in terms of typical data mining applications, suggests that is possible to get accurate brightness predictions without requiring a high number of user indications. In addition, we believe that the number of user inputs will drop down even further once CAPED has sufficiently learned the user requirements.

7. Related Work

An orthogonal area of research to our work is studying ways of improving the readability of displays at a given backlight level, rather than focusing on better predicting the best backlight level. Zhu et al. [22] provide an interesting survey of the design implications and characteristics of transfective LCD displays. Lee et al. [13] describe a hybrid display which uses either an OLED display or a transfective LCD display depending on the amount of ambient light available.

Some work which reiterates the need for user perception-based display brightness is given by Guterman et al. [7], which analyzes user preferences of display panel brightness. Their primary result is that brighter display panels are not always preferable, and that overly bright displays can be actually less preferred.

Some works focus on other ways of using context or online learning to optimize various aspects of the computing experience. Krause et al. [11] describe a method of learning user preferences of system volume from an array of biological sensors. Shye et al. [18] use an array of sensors while gradually reducing processor frequency until a significant change in the sensor readings is detected. Seshia [17] proposes a reactive system which utilizes online learning of system state to aid in error recovery. Although these works have considered user satisfaction in system-level decision making (i.e., scheduling,

dynamic voltage and frequency scaling, etc.), none of them focused on screen properties, which is the focus of our work.

8. Conclusion

In this paper, we proposed CAPED, a system which enables personalized, context-aware screen brightness predictions. We outlined the necessity of an intelligent brightness control model, and implemented our system as a userspace application. We then ran user studies to gauge the accuracy of our system's predictions. Our results showed that the manufacturer's default model for predicting user screen brightness is insufficient for many users, and showed that we can increase the mean absolute prediction accuracy by 41.9%, and improve the user's average satisfaction with the display brightness levels by 0.8 points on a 5-point scale.

As the focus of personal computing devices continues its shift of focus from raw power and specs to device usability, we believe that the user experience will be one of the most important ways for device manufacturers to differentiate themselves. That shift in manufacturer focus reflects a shift in user focus as well – a significant number of users want devices to suit them, rather than having to micromanage every aspect of their electronic devices. This increased need for automatic preference configuration and prediction lends itself well to on-device online learning systems.

We also believe that there are many other mobile subsystems which could benefit from online learning.

9. Acknowledgements

This work is supported by an Intel URO Energy Smart SoC Program grant and by NSF grants CCF-0916746 and CCF-0747201. The authors would also like to thank the user study volunteers for their participation in this study.

References

- [1] Larger screens and improved resolution drive growth in smartphone displays, according to NPD DisplaySearch. URL <http://www.prweb.com/releases/2013/6/prweb10850494.htm>.
- [2] CIE 145: The correlation of models for vision and visual performance. Standards Technical Report 145, 2002.
- [3] M. Alpern and N. Ohba. The effect of bleaching and backgrounds on pupil size. *Vision Research*, 12(5):943–951, May 1972. ISSN 0042-6989. . URL <http://www.sciencedirect.com/science/article/pii/0042698972900168>.
- [4] A. Blum. On-line algorithms in machine learning. In *In Proceedings of the Workshop on On-Line Algorithms, Dagstuhl*, pages 306–325. Springer, 1996.
- [5] D. Cline, H. W. Hofstetter, and J. R. Griffin. *Dictionary of visual science*. Butterworth-Heinemann, Boston, MA, 4th edition, 1997. ISBN 0-7506-9895-0.
- [6] J. F. Duffy, R. E. Kronauer, and C. A. Czeisler. Phase-shifting human circadian rhythms: influence of sleep timing, social contact and light exposure. *The Journal of Physiology*, 495(Pt 1):289–297, Aug. 1996. ISSN 0022-3751, 1469-7793. URL http://jp.physoc.org/content/495/Pt_1/289. PMID: 8866371.
- [7] P. S. Guterman, K. Fukuda, L. M. Wilcox, and R. S. Allison. 75.3: Is brighter always better? the effects of display and ambient luminance on preferences for digital signage. In *SID Symposium Digest of Technical Papers*, volume 41, page 11161119, 2010. URL <http://onlinelibrary.wiley.com/doi/10.1889/1.3499851/abstract>.
- [8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. In *SIGKDD Explorations*, volume 11, 2009.
- [9] H. Hashemi, M. Khabazkhoob, E. Jafarzadehpour, M. H. Emamian, M. Shariati, and A. Fotouhi. Contrast sensitivity evaluation in a population-based study in shahroud, iran. *Ophthalmology*, 119(3):541–546, Mar. 2012. ISSN 0161-6420. . URL <http://www.sciencedirect.com/science/article/pii/S0161642011007950>.
- [10] E. F. Kelley, M. Lindfors, and J. Penczek. Display daylight ambient contrast measurement methods and daylight readability. *Journal of the Society for Information Display*, 14(11):10191030, 2006. URL <http://onlinelibrary.wiley.com/doi/10.1889/1.2393026/abstract>.
- [11] A. Krause, A. Smailagic, and D. P. Siewiorek. Context-aware mobile computing: Learning context-dependent personal preferences from a wearable sensor array. *Mobile Computing, IEEE Transactions on*, 5(2):113127, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1563997.
- [12] B. M. Kudielka, I. S. Federenko, D. H. Hellhammer, and S. Wst. Morningness and eveningness: The free cortisol rise after awakening in early birds and night owls. *Biological Psychology*, 72(2):141–146, May 2006. ISSN 0301-0511. . URL <http://www.sciencedirect.com/science/article/pii/S0301051105001407>.
- [13] J.-H. Lee, X. Zhu, Y.-H. Lin, W. K. Choi, T.-C. Lin, S.-C. Hsu, H.-Y. Lin, and S.-T. Wu. High ambient-contrast-ratio display using tandem reflective liquid crystal display and organic light-emitting device. *Opt. Express*, 13(23):94319438, 2005. URL <http://lcd.creol.ucf.edu/publications/2005/Opt%20Express%20Lee%200LED.pdf>.
- [14] N. Littlestone and M. Warmuth. The weighted majority algorithm. In *30th Annual Symposium on Foundations of Computer Science, 1989*, pages 256–261, 1989. .
- [15] J. W. Miller. Study of visual acuity during the ocular pursuit of moving test objects. II. effects of direction of movement, relative movement, and illumination. *J. Opt. Soc. Am.*, 48(11):803–806, Nov. 1958. . URL <http://www.opticsinfobase.org/abstract.cfm?URI=josa-48-11-803>.
- [16] P. Ranganathan, E. Geelhoed, M. Manahan, and K. Nicholas. Energy-aware user interfaces and energy-adaptive displays. *Computer*, 39(3):3138, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1607946.
- [17] S. A. Seshia. Autonomic reactive systems via online learning. In *Proceedings of the IEEE International Conference on Autonomic Computing (ICAC)*. IEEE Press, June 2007.
- [18] A. Shye, Y. Pan, B. Scholbrock, J. S. Miller, G. Memik, P. A. Dinda, and R. P. Dick. Power to the people: Leveraging human physiological traits to control microprocessor frequency. In *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, page 188199, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4771790.
- [19] A. Shye, B. Scholbrock, and G. Memik. Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, page 168178, 2009. URL <http://dl.acm.org/citation.cfm?id=1669135>.
- [20] V. Vovk and V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56:153–173, 1997.
- [21] C. Wiley. The new generation digital display interface for embedded applications, Dec. 2010. URL <http://www.vesa.org/wp-content/uploads/2010/12/DisplayPort-DevCon-Presentation-eDP-Dec-2010-v3.pdf>.
- [22] X. Zhu, Z. Ge, T. X. Wu, and S.-T. Wu. Transflective liquid crystal displays. *Journal of Display Technology*, 1(1):15–29, Sept. 2005. ISSN 1551-319X. . URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1498782>.