This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Dynamic Backlight Scaling Optimization: A Cloud-Based Energy-Saving Service for Mobile Streaming Applications

Chun-Han Lin, *Member, IEEE,* Pi-Cheng Hsiu, *Member, IEEE,* and Cheng-Kang Hsieh

Abstract—With the increasing variety of mobile applications, reducing the energy consumption of mobile devices is a major challenge in sustaining multimedia streaming applications. This paper explores how to minimize the energy consumption of the backlight when displaying a video stream without adversely impacting the user's visual experience. First, we model the problem as a dynamic backlight scaling optimization problem. Then, we propose algorithms to solve the fundamental problem and prove the optimality in terms of energy savings. Finally, based on the algorithms, we present a cloud-based energy-saving service. We have also developed a prototype implementation integrated with existing video streaming applications to validate the practicability of the approach. The results of experiments conducted to evaluate the efficacy of the proposed approach are very encouraging and show energy savings of 15-49% on off-the-shelf mobile devices.

Index Terms—Energy-efficient optimization, dynamic backlight scaling, multimedia streaming applications, cloud services, mobile devices

1 Introduction

Advances in information and communications technology have increased the popularity of mobile devices. This in turn is motivating the development of a growing number of mobile applications and services, which are having a profound effect on people's lifestyles. However, reducing the energy consumption of mobile devices that utilize the applications and services is a major challenge. In recent decades, researchers have been exploring various low-power system designs by targeting different energy-intensive components [17, 22, 31], as well as power management policies from various perspectives [16, 24, 25]. Recent studies on mobile user activity indicated that the backlight used to illuminate the display subsystem consumes most of the energy; thus, it should receive the most attention with respect to improving energy efficiency [27, 28]. Furthermore, mobile users nowadays are becoming increasingly addicted to multimedia streaming applications, such as YouTube [8], and the ability to disseminate videos via social network communities like Facebook [4]. Such usage behavior will lead to a significant increase in the energy consumption of mobile devices, especially with the strong demand for larger, higher-resolution screens. This observation motivates us to explore how to minimize the backlight's energy consumption when browsing multimedia streaming applications on mobile devices.

- *The authors are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan. E-mail: {chunhan, pchsiu, changun}@citi.sinica.edu.tw*

A preliminary version of this paper was presented at the IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED) 2011 and won the Design Contest Award.

The display subsystem needs to stay in active mode for as long as the video stream is displayed; thus, a sensible way to reduce the energy consumption is to dim the backlight. However, this may lead to image distortion, which is normally defined as the resemblance between the original video image and the backlight-scaled image [12, 29]. For example, the *structural similarity index* [32], a metric specially designed to comply with the perception of the human eye, is widely used to assess distortion. In recent years, a number of effective *backlight scaling techniques* have been developed to limit the distortion and/or maintain the fidelity of a single image when the backlight is dimmed. In particular, the *just noticeable difference* of the *human visual system* is exploited by the approach in [21], so that the incurred image distortion is confined to a tolerable threshold and does not affect the clarity of the display significantly. Various *image compensation techniques*, e.g., [13, 14, 18], have also been proposed to further dim the backlight. They compensate for the image distortion through image pixel transformation and further reduce the energy consumption at the same time. These techniques determine the dimmest backlight level for a single image and provide a foundation for exploring dynamic backlight scaling optimization in this paper.

A video stream comprises a series of image frames. An intuitive way to reduce energy consumption is to treat a video stream as a collection of images and dynamically change the backlight by applying backlight scaling techniques to each image frame individually [10, 26]. However, in most video applications, the dimmest backlight level may vary significantly across consecutive frames [14], so changing the backlight abruptly over a number of frames may result in flickering effects and affect user perception [20]. To resolve the issue, some approaches determine the backlight level for an image frame by considering the preceding frame's pixel values and backlight level [19,20]. The drawback of this strategy is that switching the backlight level frequently may introduce inter-frame brightness distortion [12]. Furthermore, the hardware requires some time to react and adjust the backlight, so it is necessary to reduce the frequency of backlight switching [23]. To this end, the approach in [23] groups the image frames of a video and determines a common backlight level for each group. As a result, the backlight of a scene may be changed suddenly if the frames comprising the scene are partitioned into different groups. In contrast, the approach in [12] quantizes the number of backlight levels to eliminate small backlight fluctuations during a scene and, thereby prevents frequent backlight changes. The drawbacks of existing heuristics result primarily from determining the backlight level of each image frame based on only its adjacent frames (and itself), instead of having an overall consideration based on all the frames in a video. Approaches based on heuristic or empirical studies cannot provide a rigid theoretic framework for dynamic backlight scaling optimization.

In this paper, we focus on minimizing the energy consumption incurred by the backlight for multimedia streaming applications on mobile devices, without adversely impacting the user's visual experience. The contributions of this study are as follows. First, we model the problem of dynamic backlight scaling optimization that imposes three scaling constraints on the backlight changes over image frames. Second, we propose algorithms to solve the fundamental problem with different combinations of the constraints. The solution involves determining the appropriate backlight levels for image frames without violating the concerned constraints. We prove that the algorithms are optimal in terms of energy savings when the energy consumption is a strictly increasing function of the backlight levels. Third, we have deployed a *cloud-based energy-saving service* on Chunghwa Telecom (CHT) hicloud [3], where the proposed algorithms serve as the key technology for the service. We have also developed a mobile application program for Google's Android [1] and Apple's iOS [2] to validate the practicability of the approach studied in this work. When the program is installed, HTC Desire smartphones [6] and Apple iPad tablets [2] can achieve a significant energy reduction (15-29% and 34-49%, respectively) when browsing video streams on YouTube [8], but users are not aware that dynamic backlight scaling is being applied. Finally, we conducted a series of experiments and compared the proposed approach with a heuristic revised based on the approach proposed in [23]. The experimental results provide further insights into dynamic backlight scaling on mobile devices for multimedia streaming applications.

The remainder of this paper is organized as follows: Section 2 describes the system model and defines the problem. In Section 3, we propose optimal algorithms to solve the problem with different constraint combinations. In Section 4, we present a cloud-based energy-saving service and discuss technical implementation issues. The experimental results are reported in Section 5. Section 6 contains some concluding remarks.

2 System Model and Problem Definition

In this section, we present respective scaling constraints that reflect some physical characteristics of *video distortion*, *user perception*, and *hardware limitation*. Then, we introduce the power model and define the fundamental problem.

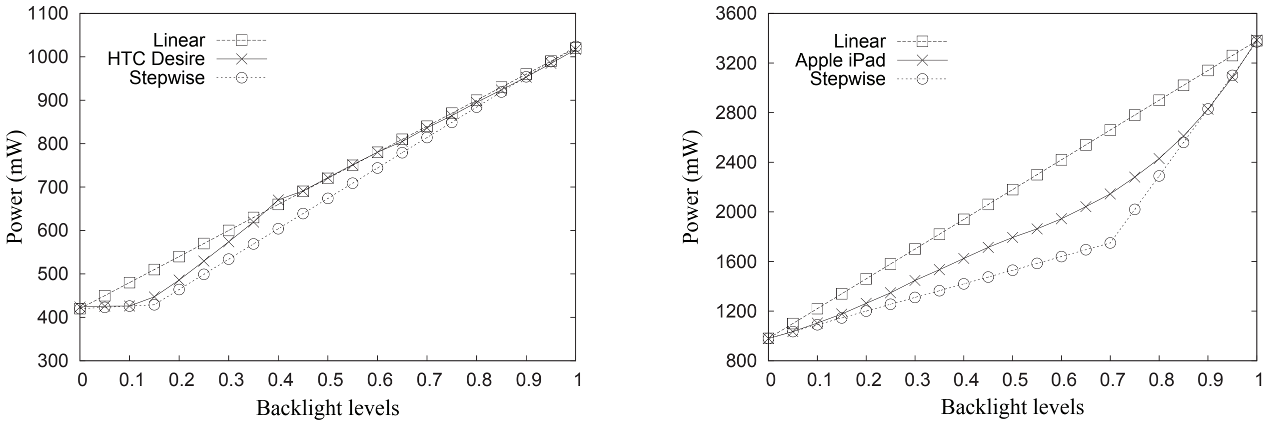
A streaming video comprises a series of N image frames, $F = \{f_1, f_2, ..., f_N\}$, displayed in succession at a constant rate. Each image frame is represented by a grid of pixels. The *perceptual luminance intensity* of a pixel shown on a display subsystem is proportional to the product of the *backlight level* and the *pixel luminance*[1] [23]. The pixel luminance does not have a noticeable impact on the energy consumption[2], but the backlight level is a decisive factor [14]. Therefore, dimming the backlight level while limiting the image distortion or compensating for the loss of the perceptual luminance intensity by increasing the pixel luminance is considered an effective way to save energy for image display on mobile devices. A number of image distortion metrics and compensation techniques have been proposed to limit image distortion and/or maintain image fidelity of a single image, e.g., [11,13,14,18,21]. In this paper, we simply assume that each image frame $f_i \in F$ is associated with a *critical backlight level*, $c(i)$, which represents its dimmest backlight level (determined by some image distortion metric or compensation technique), and treat the critical backlight level as a scaling constraint. This *distortion constraint* limits the dimmest backlight level of each image frame.

A user may perceive a wide range of stimulus magnitudes under different backlight levels. Dynamic backlight scaling is applied in a per-frame basis, and the critical backlight levels may change significantly across consecutive image frames in most videos [12,14]. The abrupt changes in backlight levels may result in an evident flickering effect, and this phenomenon will interfere with the user's visual experience of videos [12, 20]. Therefore, the maximum increase or decrease in the backlight level for a backlight change should be limited, so that the change is too subtle to be discerned by the human eye or, at least, will not incur adversely interference to the user. In the human visual system, the *just noticeable difference* is the minimum amount by which the stimulus intensity must be changed in order to produce a noticeable variation in sensory experience [20], and Weber's Law states that the ratio of the just noticeable

1. A pixel's luminance is an 8-bit value between 0 and 255. It can be derived by converting its RGB values to the YC_bC_r coordinate space with standard conversion functions [23].

2. It requires little energy to rearrange the direction of liquid crystal molecules between the alignment layers for changing the pixel transmittance [14].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



(a) HTC Desire

(b) Apple iPad

Fig. 1. Power models

difference to the stimulus intensity is a constant [9, 30]. Based on the law, we define the *differential constraint* that the increment or decrement for a backlight change is not greater than the current backlight level multiplied by a differential ratio r.

A display subsystem relies on a light source to illuminate the display panel from behind. Although the light source is usually capable of adjusting the backlight level, it may not be able to adjust the backlight promptly for every image frame, because hardware takes some time to react and usually has its limitation. In addition, frequent changes in the backlight may introduce inter-frame brightness distortion [12, 20]. Therefore, there needs to be a minimum time period (duration) between consecutive changes in backlight levels. We define another scaling constraint, referred to as the *duration constraint*, which requires the adjusted backlight level resulting from a backlight change to remain the same for a number of d subsequent image frames.

The energy consumed by a display subsystem is dominated by the backlight level as mentioned previously. Some studies simply assume that the energy consumption is linearly proportional to the backlight level and model the relationship with linear functions [11, 12]. However, as observed in [14], although the energy consumption increases with the backlight level, the increasing slope may change within different ranges of backlight levels. Various stepwise functions have been used to characterize the power models of display subsystems [18, 20, 21]. Let a display subsystem be equipped with a set of M available backlight levels $B = \{b_1, b_2, ..., b_M\}$. For ease of presentation, we map the backlight levels into corresponding dimming values, and normalize them in the range 0 to 1, with 0 representing no backlight and 1 representing the full backlight. Figure 1 shows the power models measured from an Android smartphone of HTC Desire [6] and an Apple iPad tablet [2] in practice, as well as the models approximated by linear and stepwise functions. In this paper, we make no assumption about the power model to which the proposed algorithms can be applied. However, we prove that the proposed algorithms are optimal in terms of energy savings if the power model $P()$ is a strictly increasing function of the backlight levels.

In summary, scaling down the backlight level will save energy; however, it could affect the user's visual experience adversely if the image frames are not applied with appropriate backlight levels. Therefore, our objective is to determine an appropriate backlight level for each image frame such that the total energy consumption incurred by the backlight is minimized. A mapping of a set of image frames, F, to a set of available backlight levels, B, is called a *backlight assignment*, σ, i.e., $\sigma : F \rightarrow B$. A backlight assignment is *feasible* if the three scaling constraints are satisfied: (1) the backlight level applied to any image frame is not lower than the frame's critical backlight level; (2) the magnitude of a backlight change is not greater than the current backlight level multiplied by a differential ratio; and (3) the number of image frames between any two backlight changes is not less than a specified number. Next, we formally define the fundamental problem:

The Dynamic Backlight Scaling Optimization Problem

Instance: A set of image frames $F = \{f_1, f_2, ..., f_N\}$, where each frame $f_i \in F$ is associated with a critical backlight level $c(i)$; a differential ratio r and a minimum duration of d image frames for a backlight change; and a set of available backlight levels $B = \{b_1, b_2, ..., b_M\}$ with a power model $P()$ representing the relationship between the backlight levels and the energy consumption.

Objective: A feasible backlight assignment σ such that the total energy consumption, $\sum_{i=1}^{N} P(\sigma(i))$, is minimized.

3 Dynamic Backlight Scaling Optimization

3.1 Video Distortion and User Perception

3.1.1 Algorithm Description

In this section, we propose an optimal algorithm to solve a restricted version of the dynamic backlight scaling problem. In the version, we consider only the distortion and differential constraints, and set the duration parameter d at 1. The algorithm will demonstrate the basic idea used to deal with the differential constraint when we solve the general version in a subsequent section.

ALGORITHM 1:

Input: A frame set F with critical backlight levels $c()$, a set of backlight levels B with power model $P()$, and a differential ratio r

Output: A feasible assignment σ

1 $\sigma \leftarrow c$
2 $Q \leftarrow F$
3 **while** $Q \neq \emptyset$ **do**
4 $\quad f_i \leftarrow$ remove from Q a frame with the highest level
5 $\quad$ **if** $f_{i-1} \in Q$ **then**
6 $\quad\quad \sigma[i-1] \leftarrow \max(\sigma[i-1], \mathtt{MinLv}(\frac{\sigma[i]}{1+r}))$
7 $\quad$ **if** $f_{i+1} \in Q$ **then**
8 $\quad\quad \sigma[i+1] \leftarrow \max(\sigma[i+1], \mathtt{MinLv}(\sigma[i] \times (1-r)))$
9 **return** σ

Given a frame set F with critical backlight levels $c()$, a set of backlight levels B with power model $P()$, and a differential ratio r, Algorithm 1 determines a backlight assignment σ without violating the two scaling constraints. At the beginning, each frame is initially assigned with its critical backlight level (Line 1). Throughout the algorithm, we maintain a priority queue Q that initially contains all the frames in F, keyed by their current backlight levels (Line 2). We repeatedly remove from Q a frame f_i with the highest backlight level until Q is empty (Lines 3-4). Whenever frame f_i is extracted, its backlight level is determined and will never change. Then, the two adjacent frames, f_{i-1} and f_{i+1}, are examined whether their backlight levels should be updated, so as to satisfy the differential constraint, as shown by the example in Figure 2. If f_{i-1} is still in Q (Line 5), its current backlight level, $\sigma[i-1]$, must be lower than or equal to the level, $\sigma[i]$, assigned to f_i. Since the increment of a backlight change cannot be greater than the current backlight level multiplied by the differential ratio r, frame f_{i-1} should be assigned a level at least $\frac{\sigma[i]}{1+r}$. Let $\mathtt{MinLv}(x)$ denote the minimum available level that is not lower than x, i.e., $\mathtt{MinLv}(x) = \min\{b_j \geq x | \ \forall\ b_j \in B\}$. The level assigned to f_{i-1} should be updated if the current level $\sigma[i-1]$ is lower than $\mathtt{MinLv}(\frac{\sigma[i]}{1+r})$ (Line 6). Similarly, if f_{i+1} is in Q, since the decrement of a backlight change has to be bounded, the currently assigned level $\sigma[i+1]$ is updated to $\mathtt{MinLv}(\sigma[i] \times (1-r))$ if necessary (Lines 7-8). At the end, the backlight assignment σ is returned (Line 9).

3.1.2 The Properties of Algorithm 1

For the rest of this section, we analyze the time complexity of Algorithm 1 and prove its optimality for the restricted version.

Lemma 1: The time complexity of Algorithm 1 is $O(N \ln NM)$.

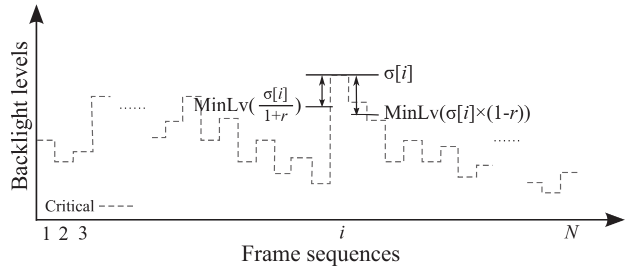


Fig. 2. An example to illustrate Algorithm 1

Proof. This analysis assumes that the priority queue Q and the function $\mathtt{MinLv}()$ are implemented by a binary heap and a binary search algorithm, respectively [15]. The initialization of array σ and the building of heap Q based on F in Lines 1-2 can be done in $O(N)$ time. The **while** loop in Lines 3-8 is exactly executed N times. Within the loop, each operation on Q and each call to $\mathtt{MinLv}()$ cost time $O(\ln N)$ and $O(\ln M)$, respectively. Note that each update to σ in Line 6 or Line 8 also involves an implicit operation on the heap to maintain the heap property. Each loop contributes $O(\ln N + \ln M) = O(\ln NM)$ to the running time. Thus, the time complexity of Algorithm 1 is $O(N \ln NM)$. ∎

Theorem 1: Algorithm 1 is an optimal algorithm for the dynamic backlight scaling problem when only the distortion and differential constraints are considered.

Proof. Each frame's backlight level is initialized as its critical backlight level and will never decrease, so σ satisfies the distortion constraint. Moreover, the backlight levels between adjacent frames are limited to the differential constraint, because the removal of a frame from Q causes at most two frames to increase their backlight levels, and the adjusted levels must never be higher than the determined levels for the frames that have been removed from Q. Thus, σ is a feasible assignment.

Next, we prove its optimality by contradiction. Suppose there exists a feasible assignment σ' with energy consumption $\sum_{i=1}^{N} P(\sigma'[i]) < \sum_{i=1}^{N} P(\sigma[i])$. Since $P()$ is a strictly increasing function, there must be at least one frame whose backlight level in σ' is strictly lower than in σ. Let f_k be the first frame with $\sigma'[k] < \sigma[k]$ during the execution of Algorithm 1. At the time instance immediately before f_k is removed from Q, we should have $\sigma'[k] < \sigma[k]$ and $\sigma'[i] \geq \sigma[i]$, $\forall f_i \notin Q$. We delineate four possible cases, depending on whether $f_{k-1}, f_{k+1} \in Q$.

1) If $f_{k-1}, f_{k+1} \in Q$, the two frames have not been removed, so the backlight level of f_k must have not been updated yet. Thus, $\sigma'[k] < \sigma[k] = c(k)$.
2) If $f_{k-1} \notin Q$ and $f_{k+1} \in Q$, when f_{k-1} was removed, the backlight level of f_k may be updated. That is, $\sigma'[k] < \sigma[k] = \max(\mathtt{MinLv}(\sigma[k-1] \times (1-r)), c[k]) \leq \max(\mathtt{MinLv}(\sigma'[k-1] \times (1-r)), c[k])$.
3) If $f_{k-1} \in Q$ and $f_{k+1} \notin Q$, when f_{k+1} was removed, the backlight level of f_k may be updated.

$$E(i) = \begin{cases} i \times P(\max\limits_{1 \le k \le i} c(k)), & \text{if } 1 \le i \le d; \\ \min\limits_{\max(1, i-2d) \le j \le i-d} \{E(j) + \texttt{NumFm}(i,j) \times P(\max\limits_{j < k \le i} c(k))\}, & \text{otherwise.} \end{cases} \quad (1)$$

That is, $\sigma'[k] < \sigma[k] = \max(c[k], \texttt{MinLv}(\frac{\sigma[k+1]}{1+r})) \le \max(c[k], \texttt{MinLv}(\frac{\sigma'[k+1]}{1+r}))$.

4) If $f_{k-1}, f_{k+1} \notin Q$, by similar arguments in the above cases, $\sigma'[k] < \max(\texttt{MinLv}(\sigma'[k-1] \times (1-r)), c[k], \texttt{MinLv}(\frac{\sigma'[k+1]}{1+r}))$.

To conclude, in all the cases, $\sigma'[k]$ violates either the distortion constraint or the differential constraint, which contradicts the assumption that σ' is a feasible assignment. Thus, the theorem follows. ∎

3.2 Video Distortion and Hardware Limitation

3.2.1 Algorithm Description

In this section, we present a dynamic-programming algorithm and its polynomial-time implementation to solve another restricted version of the dynamic backlight scaling problem. In this version, we consider only the distortion and duration constraints, and set the differential ratio r at ∞[3]. The algorithm will demonstrate the basic idea of how we deal with the duration constraint, as used in a subsequent section to solve the general version.

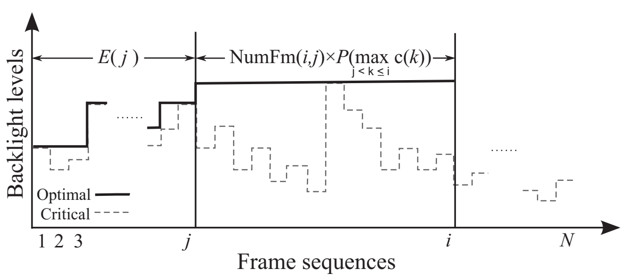


Fig. 3. An illustration to the dynamic-programming formula in Equation (1)

The algorithm is based on the recursive formula given in Equation (1). Let $E(i)$ be the minimum energy required to display first i frames in F without violating the two scaling constraints, provided that a backlight change is allowed at the subsequent frame f_{i+1}. For ease of presentation, we introduce *blank frames* and append them to F. The critical backlight level of each blank frame is set at 0, i.e., $c(i) = 0, \forall f_i > N$. It is assumed that the blank frames do not consume any energy regardless of the backlight levels applied to them. We delineate two possible cases in Equation (1):

1) If $1 \le i \le d$, then $E(i)$ is set at $i \times P(\max\limits_{1 \le k \le i} c(k))$. That is, the i frames are assigned a common backlight level, which is equal to the maximum of their critical backlight levels.

2) Otherwise, if $i > d$, suppose that the last $i - j$ frames are assigned the same backlight level, where $i > j \ge 1$. Because any frame f_k for all $k > N$ is a blank frame that does not consume any energy, we should only consider the non-blank frames when computing the energy consumption of the $i - j$ frames. Let $\texttt{NumFm}(i,j)$ denote the number of non-blank frames from f_i to f_{j+1}, i.e., $\texttt{NumFm}(i,j) = \min(i, N) - \min(j, N)$. By definition, the minimum energy consumption of the first j frames is $E(j)$, provided that a backlight change is allowed at frame f_{j+1}. Thus, the total energy consumption is $E(j) + \texttt{NumFm}(i,j) \times P(\max\limits_{j<k\le i} c(k))$, as shown by the example in Figure 3. Note that after a backlight change, the backlight level must remain the same for at least d frames. To ensure that the backlight can be changed at frame f_{i+1}, the j value should be in the range 1 to $i - d$. However, we do not consider all the $i - d$ possible values. Instead, by considering at most $d + 1$ possible values for j in the range $\max(1, i - 2d)$ to $i - d$, $E(i)$ is set as the minimum energy derived.

The objective is to derive $E(N + d)$. Note that the last d blank frames consume no energy, and are introduced to relax the presupposition that the last d non-blank frames of the N frames are subject to the same backlight level. Consequently, $E(N + d)$ is equal to the minimum energy required to display the original N frames without violating the distortion and duration constraints.

ALGORITHM 2:

Input: A frame set F with critical backlight levels $c()$, a set of backlight levels B with power model $P()$, and a minimum duration d

Output: The energy consumption for a feasible assignment σ

for $i \leftarrow 1$ **to** $N + d$ **do**
  $T[i] \leftarrow \infty$
return $E(N + d)$

Procedure $E(i)$

if $T[i] < \infty$ **then**
  return $T[i]$
if $1 \le i \le d$ **then**
  $T[i] \leftarrow i \times P(\max\limits_{1 \le k \le i} c(k))$
else
  $T[i] \leftarrow \min\limits_{\max(1,i-2d) \le j \le i-d} \{E(j) + \texttt{NumFm}(i,j) \times P(\max\limits_{j<k\le i} c(k))\}$
return $T[i]$

Algorithm 2 implements the dynamic-programming formula in Equation (1) recursively. Once derived, the

3. For ease of presentation, we assume $0 \times \infty = \infty$ in this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

solution to each subproblem $E(i)$ is stored in a corresponding entry $T[i]$ in a one-dimensional table T. At the beginning of the algorithm, each table entry is initialized as ∞ to indicate that the corresponding subproblem has not been solved yet. Whenever Procedure $E(i)$ is invoked, the procedure simply returns the previously derived solution stored in entry $T[i]$ if the entry has been updated. Otherwise, the solution to the subproblem is derived based on the dynamic-programming formula in Equation (1) and returned.

After the entire table T has been derived, a corresponding feasible assignment σ can be constructed by back tracing the table based on the dynamic-programming formula as follows. We begin with the last entry by setting an index i as $N+d$, and examine each entry backward sequentially from $T[i-d]$ to $T[i-2d]$, until we find an entry $T[j]$ in which the stored value minimizes the solution to subproblem $E(i)$. The $i-j$ frames, $f_{j+1}, f_{j+2}, ..., f_i$, are assigned the same backlight level $\max_{j<k\leq i} c(k)$. We then start with the discovered entry by setting i as j and repeat the above process recursively until $i \leq d$. Finally, we assign the remaining i frames, $f_1, f_2, ..., f_i$, the same backlight level $\max_{1\leq k\leq i} c(k)$. Because we have to examine N table entries at most, and each examination takes constant time $O(1)$ according to the time complexity analysis in Lemma 2, the construction of a feasible assignment σ based on table T can be completed in $O(N)$ time.

3.2.2 The Properties of Algorithm 2

In the remainder of this section, we analyze the time complexity of Algorithm 2 and prove its optimality for this restricted version.

Lemma 2: The time complexity of Algorithm 2 is $O(dN)$.

Proof. It is assumed that $d < N$; otherwise, the trivial case is solvable in $O(N)$ time. The time complexity of the algorithm depends on the number of table entries and the time required to derive the solution to a subproblem. The table contains $O(N)$ entries, each of which is initialized and then updated as the solution to the corresponding subproblem once. The solution to each subproblem $E(i)$ is derived by referring to $d+1$ preceding entries sequentially. When an entry $T[j]$ is referred to, deriving a candidate solution to $E(i)$ takes $O(1)$ time. Note that $\max_{j<k\leq i} c(k)$ can be computed in constant time $O(1)$ by simply comparing $c(j+1)$ and $\max_{j+1<k\leq i} c(k)$, because the latter term has been computed when entry $T[j+1]$ was referred to. Therefore, deriving the solution to a subproblem takes $O(d)$ time. In summary, table T can be constructed in $O(dN)$ time. ∎

Lemma 3: In Equation (1), when $i > d$, considering $\max(1, i-2d) \leq j \leq i-d$ is equivalent to considering all the possible values for j in the range 1 to $i-d$.

Proof. If $d < i \leq 2d+1$, then $\max(1, i-2d) = 1$, and this lemma obviously holds. For $i > 2d+1$, we prove this lemma by showing that all the candidate solutions derived for $E(i)$ when $1 \leq j < i-2d$ are not smaller than the candidate solution derived when $j = i-d$. That is, if we can prove that $\forall\ 1 \leq j < i-2d$,

$$\begin{aligned}&E(j) + \texttt{NumFm}(i,j) \times P(\max_{j<k\leq i} c(k))\\ &\geq E(i-d) + \texttt{NumFm}(i,i-d) \times P(\max_{i-d<k\leq i} c(k)),\end{aligned}$$

then there is no need to consider $1 \leq j < i-2d$ when we derive the minimum solution for $E(i)$.

Without loss of generality, let us consider any j, where $1 \leq j < i-2d$. Because the maximum of a set is never smaller than the maximum of any subset, we can derive the following inequality by partitioning the range $(j,i]$ into two subranges $(j, i-d]$ and $(i-d, i]$.

$$\begin{aligned}&E(j) + \texttt{NumFm}(i,j) \times P(\max_{j<k\leq i} c(k))\\ &\geq E(j) + \texttt{NumFm}(i-d,j) \times P(\max_{j<k\leq i-d} c(k))\\ &\qquad + \texttt{NumFm}(i,i-d) \times P(\max_{i-d<k\leq i} c(k))\end{aligned}$$

Because $j < i-2d$, $\texttt{NumFm}(i-d,j) \geq d$, which implies that a backlight change is allowed at frame f_{i-d+1}. Furthermore, the energy consumption of any backlight assignment for the first $i-d$ frames, provided that a backlight change is allowed at frame f_{i-d+1}, is no smaller than the minimum energy consumption $E(i-d)$, so we have

$$\begin{aligned}&E(j) + \texttt{NumFm}(i-d,j) \times P(\max_{j<k\leq i-d} c(k))\\ &\geq E(i-d).\end{aligned}$$

By comparing the above two inequalities, the lemma follows. ∎

Theorem 2: Algorithm 2 is an optimal algorithm for the dynamic backlight scaling problem when only the distortion and duration constraints are considered.

Proof. The theorem follows directly from the correctness of the dynamic-programming formula $E(i)$ in Equation (1). We prove its correctness by mathematical induction on the index i. As the induction basis, when $1 \leq i \leq d$, to ensure that a backlight change is allowed at frame f_{i+1}, the first i frames must be assigned the same backlight level. Since $P()$ is a strictly increasing function, the minimum energy consumption required to display the i frames is $i \times P(\max_{1\leq k\leq i} c(k))$. Thus, the formula is correct. For the induction hypothesis, suppose that the formula is always correct for the first i frames when $i < n$. We show that the formula is also correct for the first n frames.

Suppose that the last $n-j$ frames are assigned a common backlight level. Then, the minimum energy consumption required to display the $n-j$ frames is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

$$E_{s,\underline{b}}^{t,\overline{b}} = \begin{cases} 0, & \text{if } t = s; \\ \infty, & \text{else if } t < s + d; \\ \min\limits_{\substack{\max(s,h-2d) \le j < h, \\ \max(h,j+d) \le i \le \min(h+2d,t)}} \{E_{s,\underline{b}}^{j,\mathtt{MinLv}(\frac{\hat{b}}{1+r})} + \mathtt{NumFm}(i,j) \times P(\hat{b}) + E_{i,\mathtt{MinLv}(\hat{b}(1-r))}^{t,\overline{b}}\}, & \text{otherwise.} \end{cases} \quad (2)$$

$\mathtt{NumFm}(n,j) \times P(\max\limits_{j<k\le n} c(k))$. Since $j < n$, by the induction hypothesis, $E(j)$ is the minimum energy consumption for the first j frames, provided that a backlight change is allowed at the subsequent frame f_{j+1}. Since the common backlight level applied to the last $n-j$ frames is not subject to the level assigned to frame f_j, the total energy consumption of the n frames is the sum of the energy consumption of the first j frames and that of the last $n-j$ frames. To allow a backlight change at frame f_{n+1}, we must ensure that $n-j \ge d$. This implies that all possible values for j are in the range 1 to $n-d$. Based on Lemma 3, considering $\max(1, n-2d) \le j \le n-d$ is sufficient to derive the minimum among all the $n-d$ candidate solutions. Hence, $E(n)$ is the minimum energy consumption of the n frames, provided that a backlight change is allowed at frame f_{n+1}. The theorem follows. ∎

3.3 Video Distortion, User Perception, and Hardware Limitation

3.3.1 Algorithm Description

In this section, we present a dynamic-programming algorithm and its polynomial-time implementation to solve the general version of the dynamic backlight scaling problem.

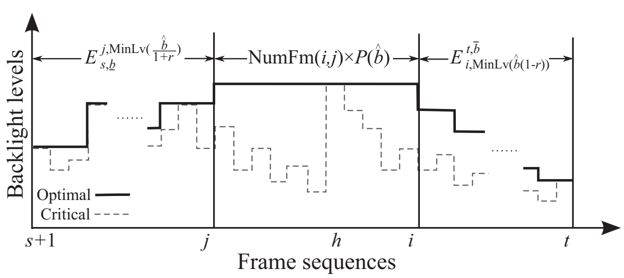


Fig. 4. An illustration to the dynamic-programming formula in Equation (2)

Let $E_{s,\underline{b}}^{t,\overline{b}}$ be the minimum energy required to display the frames in the range $(f_s, f_t]$ without violating the three scaling constraints, provided that the following conditions are satisfied: (1) the level of f_{s+1} is not lower than $\underline{b}$; (2) the level of f_t is not lower than $\overline{b}$; and (3) backlight changes are allowed at both f_{s+1} and f_{t+1}. Consequently, f_{s+1} and f_t should be assigned levels not lower than $\max(c(s+1), \underline{b})$ and $\max(c(t), \overline{b})$ respectively, while any other frame f_k should be assigned a level not lower than $c(k)$. Let f_h be a frame which should be assigned a level not lower than $\hat{b}$, where $\hat{b} = \max(\max\limits_{s<k\le t} c(k), \underline{b}, \overline{b})$. For ease of presentation, we add d blank frames in the front and at the back of F, respectively. We delineate three possible cases in Equation (2):

1) If $t = s$, then $E_{s,\underline{b}}^{t,\overline{b}}$ is set at 0. That is, the frame set is empty and consumes no energy.
2) If $s < t < s+d$, then $E_{s,\underline{b}}^{t,\overline{b}}$ is set at ∞. That is, the frame set contains at most $d-1$ frames, so the energy consumption is set at ∞ to indicate there is no feasible assignment in which backlight changes are allowed at f_{s+1} and f_{t+1}.
3) Otherwise, $t \ge s+d$. Suppose that the frame f_h is in some range $(f_j, f_i]$, where the $j-i$ frames are assigned the same level $\hat{b}$. Then, the energy consumption of these frames is $\mathtt{NumFm}(i,j) \times P(\hat{b})$. For the frames in the front range $(f_s, f_j]$, to ensure that f_{j+1} can be assigned the level $\hat{b}$, f_j should be assigned a level not lower than $\mathtt{MinLv}(\frac{\hat{b}}{1+r})$. Thus, the energy consumption for these frames is $E_{s,\underline{b}}^{j,\mathtt{MinLv}(\frac{\hat{b}}{1+r})}$. Similarly, the energy consumption for the rear range $(f_i, f_t]$ is $E_{i,\mathtt{MinLv}(\hat{b}(1-r))}^{t,\overline{b}}$. If a backlight change is allowed at f_{i+1}, then the total energy consumption is the sum of the energy consumption for the three ranges, as shown by the example in Figure 4. The possible values for j are in the range $[s, h)$. To ensure that a backlight change is allowed at f_{i+1}, we should have $i \ge j+d$, so the possible values for i are in the range $[\max(h, j+d), t]$. However, we do not consider all the possible pairs of j and i. Instead, by considering at most $2d$ values for j in the range $[\max(s, h-2d), h)$ and at most $2d+1$ values for i in the range $[\max(h, j+d), \min(h+2d, t)]$, $E_{s,\underline{b}}^{t,\overline{b}}$ is set as the minimum energy derived.

The objective is to derive $E_{-d,0}^{N+d,0}$. Note that the d front (resp. rear) blank frames are introduced to relax the presupposition that the first (resp. last) d non-blank frames of the N original frames are subject to the same backlight level. The formula in Equation (2) can be implemented by a recursive algorithm, referred to as Algorithm 3, in a similar way to the implementation of Algorithm 2, except that Algorithm 3 employs a four-dimensional table[4] $T[s, t, \underline{b}, \overline{b}]$ with each entry initialized as $-\infty$ to indicate that the subproblem has not been solved yet.

4. For ease of presentation, we allow that table entries have negative index values. In practice, all the index values are shifted with the same constant.

3.3.2 The Properties of Algorithm 3

In the remainder of this section, we analyze the time complexity of Algorithm 3 and prove its optimality for the dynamic backlight scaling optimization problem.

Lemma 4: The time complexity of Algorithm 3 is $O(N^2M^2(N + \ln M + d^2))$.

Proof.
It is assumed that $d < N$; otherwise, the trivial case is solvable in $O(N)$ time. The time complexity depends on the number of table entries and the time required to derive the solution to a subproblem. For an entry $T[s, t, \underline{b}, \overline{b}]$, the first two indexes have $O(N)$ possible values and the last two have $O(M)$ possible values; thus, the table contains $O(N^2M^2)$ entries. Each entry is initialized and stored with the solution to the corresponding subproblem at most once. The solution to a subproblem can be derived in $O(N + \ln M + d^2)$ time by three steps: (1) finding the frame f_h and the level $\hat{b}$ takes $O(N)$ time; (2) the two calls to $\mathtt{MinLv}()$ cost $O(\ln M)$ time; and (3) considering each of the $O(d^2)$ pairs of j and i derives the minimum, where a candidate solution with respect to each pair costs $O(1)$ time. ∎

Lemma 5: In Equation (2), when $t \geq s + d$, considering $j \in [\max(s, h-2d), h)$ and $i \in [\max(h, j+d), \min(h+2d, t)]$ is equivalent to considering all the possible pairs of $j \in [s, h)$ and $i \in [\max(h, j + d), t]$.

Proof. This lemma can be proved in a similar way to the proof of Lemma 3 by showing two arguments. First, all the candidate solutions derived for $E^{t,\overline{b}}_{s,\underline{b}}$ when $j \in [s, h-2d)$ are not smaller than the solution derived when $j = h-d$; thus, there is no need to consider $j \in [s, h-2d)$ when we derive the minimum solution for $E^{t,\overline{b}}_{s,\underline{b}}$. Then, all the candidate solutions derived when $i \in (h + 2d, t]$ are not smaller than the solution derived when $i = h+d$; thus, there is no need to consider $i \in (h + 2d, t]$.

Theorem 3: Algorithm 3 is an optimal algorithm for the dynamic backlight scaling optimization problem.

Proof. We prove this theorem by mathematical induction on the size, i.e., the number of frames, of $(f_s, f_t]$ in Equation (2). Two cases are proved as the induction basis. If $t = s$, the energy consumption is 0 since the frame set is empty. If $s < t < s+d$, there is no feasible assignment in which backlight changes are allowed at both f_{s+1} and f_{t+1}, so the energy consumption is deemed to be ∞. For the induction hypothesis, suppose that the formula is correct for any frame set of size smaller than n. We show that the formula is also correct for a frame set of size n.

Let $(f_j, f_i]$ be any frame subset (of the frame set $(f_s, f_t]$) which includes the frame f_h and comprises frames assigned the same backlight level. Since the level assigned to f_h should not be lower than $\hat{b}$, the minimum energy required for the subset is $\mathtt{NumFm}(i, j) \times P(\hat{b})$. For the front frame subset $(f_s, f_j]$, to ensure that f_{j+1} can be assigned $\hat{b}$, frame f_j should be assigned a level not lower than $\mathtt{MinLv}(\frac{\hat{b}}{1+r})$. Because the size of the front subset is smaller than n, by the induction hypothesis, the minimum energy required for the subset is $E^{j,\mathtt{MinLv}(\frac{\hat{b}}{1+r})}_{s,\underline{b}}$, provided that a backlight change is allowed at f_{j+1}. Similarly, the minimum energy required for the rear subset $(f_i, f_t]$ is $E^{t,\overline{b}}_{i,\mathtt{MinLv}(\hat{b}(1-r))}$, provided that a backlight change is allowed at f_{t+1}. If a backlight change is allowed at f_{i+1}, then the total energy required for $(f_s, f_t]$ is the sum of that required for the three subsets. To allow a backlight change at f_{i+1}, possible values for j and i are in the ranges $[s, h)$ and $[\max(h, j+d), t]$, respectively. Based on Lemma 5, the ranges of j and i can be reduced to $[\max(s, h - 2d), h)$ and $[\max(h, i + d), \min(h + 2d, t)]$, respectively. By considering the possible pairs of j and i, $H^{t,\overline{b}}_{s,\underline{b}}$ is set as the minimum energy derived, and the theorem follows. ∎

4 A Cloud-Based Energy-Saving Service

4.1 System Design and Implementation

In this section, we present a cloud-based energy-saving service, called the *dynamic backlight scaling service*, which minimizes the backlight's energy consumption when displaying video streams on mobile devices. A possible way to realize the service would be to develop it as a value-added service offered by Internet service providers. Then, mobile users could apply for the energy-saving service in the same way that they apply for other value-added services, such as the short message service. The service is presented in a way that is easy for end users to understand. They do not need to know how the service is provided and where the system that delivers the service is located. In the following, we present the design concepts and discuss in detain the implementation issues that arise on the cloud side and the mobile device side.

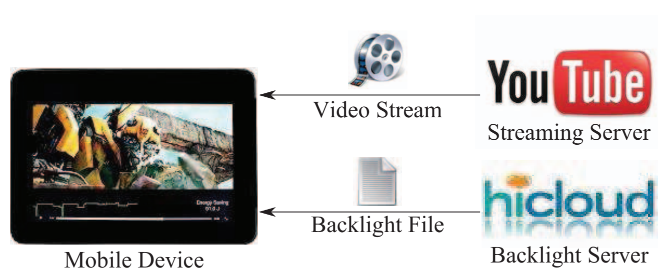


Fig. 5. The system architecture

We have deployed the dynamic backlight scaling service on CHT hicloud [3]. The prototype system, which is integrated with existing video streaming services, includes an on-line cloud server and a mobile application program developed for Google's Android [1] and Apple's iOS [2], as shown in Figure 5. The cloud server, called the *backlight server*, uses the proposed algorithms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

to analyze the videos on major video streaming websites like YouTube [8]. Note that determining the backlight levels for the image frames of a video involves analyzing a large number of image pixels and is therefore computationally intensive. The designated cloud server is responsible for running the algorithms and determining the backlight assignments, as well as for exempting mobile devices from the computational overheads. Each derived backlight assignment is stored in a text file, called the *backlight file*, in a space-efficient format and associated with the corresponding video's URL link. Because the size of a backlight file for a 15MB video (with a bit rate of 550-650 kbps) is usually less than 1 KB, it can be transmitted quickly when the wireless bandwidth is sufficient for video streaming.

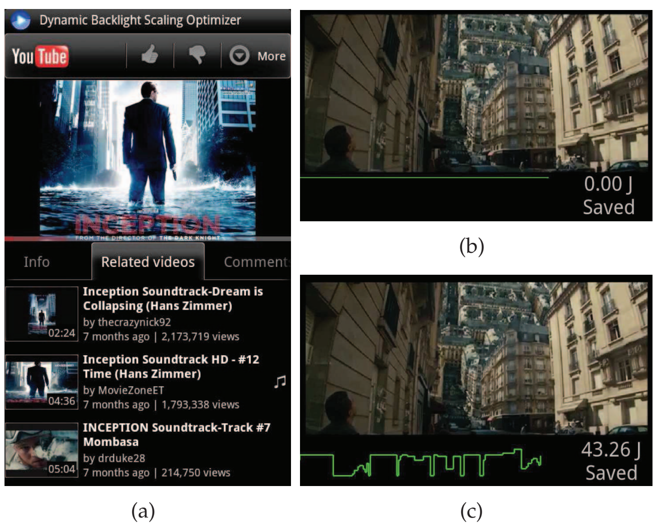


Fig. 6. Snapshots of the mobile application program

It would be ideal if all the videos on streaming websites could be analyzed in advance, but doing so would be tremendously time-consuming. Hence, we analyzed the most popular videos on YouTube, and left the remainder to be analyzed on demand. The on-demand design is inspired by an observation that the majority of videos on streaming websites are of little interest to most people and the popularity of frequently viewed videos usually varies over time. To support on-demand video analysis, we developed an application program with two versions that run respectively on Android and iOS mobile devices, as shown in Figure 6(a). When a user starts to play a video stream on a mobile device with the program installed, the program sends the video's URL link to the backlight server. If there is no corresponding backlight file, the server replies accordingly and starts analyzing the video to generate the corresponding file. In this case, the video is played without dynamic backlight scaling, as shown in Figure 6(b). Conversely, if a corresponding backlight file exists, the server returns it and the program adjusts the backlight dynamically according to the file with the video being played on the mobile device, as shown in Figure 6(c). With the on-demand design, the first user browses the video stream as usual, but enables the backlight server to analyze the video so that the subsequent users browsing the same video can benefit from the dynamic backlight scaling service.

4.2 Issues on the Cloud Side

The backlight server is responsible for generating the video stream's backlight file when requested by a mobile device. Because different mobile devices may be equipped with different display subsystems and consequently power models, we divide the generation process into two phases, *device-independent* and *device-dependent*, for optimization purposes. The device-independent phase analyzes the video stream, and determines each image frame's critical backlight level, which represents the dimmest backlight level at which the incurred image distortion does not affect the clarity of the display significantly. Based on the levels, the device-dependent phase then executes one of the proposed algorithms to derive an optimal backlight assignment for the mobile device.

In the device-independent phase, any image distortion metric that estimates the similarity between the original video image and the backlight-scaled image could be used to compute a video's critical backlight levels. An image compensation technique could also be used to further lower the critical backlight levels and potentially improve the energy savings. However, when an image compensation technique is applied, the video player's decoder must be modified to increase the pixel luminance when playing videos. For the compatibility with popular players, a simple image distortion metric was employed. In the system prototype, we utilized the structural similarity (SSIM) index [32], a metric specially designed to comply with the perception of the human eye and widely used to assess distortion. The resultant SSIM index is a decimal value between -1 and 1, where the value 1 is only achievable in the case of two identical sets of data. Given a video stream, the critical backlight level of each image frame is computed with respect to a specified SSIM index. The time complexity of this phase depends on the adopted metric and is $O(NP^3 \log M)$, where N, P, and M, denote the number of image frames, the number of pixels per frame, and the number of available backlight levels, respectively. Note that a video stream's critical backlight levels are unrelated to mobile devices and only computed once. In addition, each image frame can be analyzed independently and concurrently, so a cloud server with hundreds of virtual cores is especially well suited to the high-parallelism, computationally-intensive task.

In the second phase, one of the proposed algorithms is used to determine the optimal backlight assignment with respect to the mobile device. This phase is device-dependent because, in addition to the critical backlight levels, the algorithm takes two device-dependent parameters as its input: the mobile device's power model $P()$ and minimum applicable duration d between two adjacent backlight changes. The output is a backlight assignment for the video stream and the device model.

A feasible assignment satisfies some or all of the scaling constraints, depending on the device's characteristics and the desired visual quality. The time complexity of this phase depends on the adopted algorithm and is $O(N \ln NM)$, $O(dN)$, or $O(N^2M^2(N + \ln M + d^2))$, as analyzed in Section 3. The assignment is stored in a file specifying the time instants (relative to the beginning of the video) when the backlight should be changed and the levels that should be used. The idea is similar to that of subtitles in a video. The representation allows a video stream's backlight file to be portable for various video formats.

4.3 Issues on the Device Side

The dynamic backlight scaling service is not intended for all models of mobile devices. It is designed for popular models and those that need the service, e.g., HTC Desire [6] and Apple iPad [2]. Optimizing a mobile device's energy consumption must rely on the information about the device's power model. However, the accuracy of the power model will only affect the amount of energy saved, not the user's visual experience; therefore, other mobile devices could also benefit from the service even if their accurate power models have not be acquired. In this section, we explain how to acquire the power model of a mobile device's display subsystem, and discuss the design concept behind the mobile application program.

In practice, we use the Power Monitor produced by Monsoon Solutions [7] to measure the power models of display subsystems. Under the prototype system, we divide the luminance range into 21 equal backlight levels. We then map and normalize the levels into 21 dimming values in the range 0 to 1, where 0 represents no backlight and 1 represents the full backlight. The reason behind the setting is that it was difficult to determine whether the backlight has been changed when we increased the dimming value by 0.05 at a time. For example, Figure 1 shows the power model measured from HTC Desire, which is equipped with a 3.7-inch Super LCD display subsystem illuminated by a cold cathode fluorescent lamp, as well as the power model measured from Apple iPad, which is equipped with a 9.7-inch Multi-Touch IPS display subsystem illuminated by a light-emitting diode. In addition, HTC Desire allows backlight changes up to 8 times per second, which corresponds to a minimum duration of 4 frames for 30 frame-per-second videos, while Apple iPad allows more than 30 backlight changes in one second.

The mobile application program is developed as a resident daemon running in the background. It is activated at the same time when the *codec* is invoked by any video player. The codec is the underlying program that encodes/decodes video streams. When the daemon is activated, it sends the mobile device's model and the video's URL link to the backlight server via the HTTP protocol. On receipt of the request, the backlight server checks whether the device model supports the dynamic backlight scaling service and whether the video's backlight file has been generated. If the backlight file is received, the daemon creates a control thread to adjust the backlight dynamically with the video being played. The thread synchronizes with the codec periodically to support the Pause, Fast Forward, and Fast Backward functions. With such a design[5], no matter which application programs (such as YouTube App or Web Browser) mobile users employ to browse various video stream formats (such as MPEG4 or H.264) on different websites, they could benefit from the energy-saving service without changing their user preferences. In addition, it is intuitive that the energy savings increase as the SSIM index (video quality) decreases. The program allows users to adjust the SSIM index in order to arrange a tradeoff between energy savings and video quality. As shown in Figure 6, the program also plots the backlight changes in real time for users' reference, and estimates the accumulated energy savings (in joules or percentage) based on the mobile device's power model.

5 Real-World Case Studies

5.1 Experimental Setup

To better understand the properties of, and gain insights into, dynamic backlight scaling for mobile streaming applications, we performed experiments on some real-world video streams to validate the practicability of our approach and evaluate the performance of Algorithm 3 proposed for the general version of our problem.

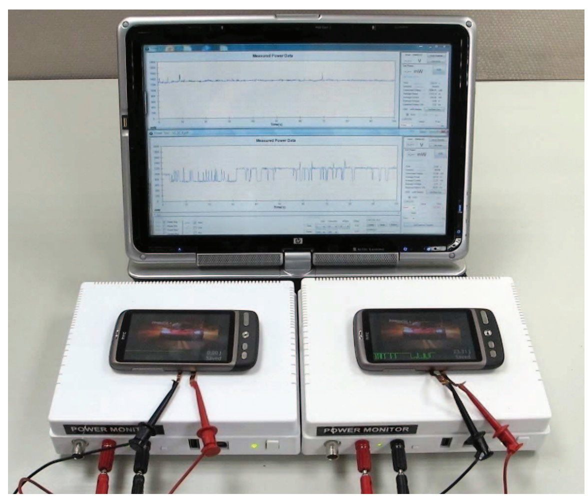


Fig. 7. The experimental environment

The dynamic backlight scaling service was deployed on CHT hicloud. The system resources allocated for the service are two 64-bit virtual cores, 4GB memory, 100GB storage capacity, and the maximum network bandwidth of 1Gbps. With such resources, the backlight server requires some time, ranging from dozens of minutes to a few hours, to generate a backlight file for a video stream, depending on the video's characteristics and the

5. This design has been fully realized in the version developed for Android. For iOS, users need to install our video player and jailbreak their iPad tablets, because programming on iOS is subject to some restrictions.

algorithm's input settings. Nevertheless, if the backlight file has been compiled, it takes only hundreds of milliseconds to obtain the file from the backlight server via 54Mbps Wi-Fi. The mobile application program was installed on two HTC Desire smartphones [6] and two Apple iPad tablets [2]. In the experimental environment, as shown in Figure 7, one smartphone/tablet performed dynamic backlight scaling while the other did not, and the transient power of the mobile devices was measured by Power Monitors of Monsoon Solutions [7].

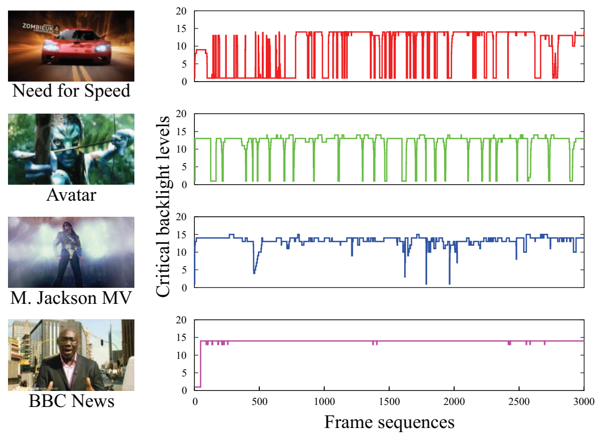


Fig. 8. Snapshots and critical backlight levels of four videos

We studied four videos with different characteristics, namely, Need for Speed, Avatar, M. Jackson MV, and BBC News, all of which can be found on YouTube. Figure 8 shows some snapshots of the videos and the critical backlight levels when the SSIM index was set at 0.9. Need for Speed, which is a typical advertisement for video games, comprises diverse scenes that demonstrate the game's high fidelity; thus, the critical backlight levels vary significantly. Avatar is a characteristic trailer for sci-fi movies, so the high fidelity feature is also emphasized, but the consecutive scenes are longer and more homogeneous than those of game advertisements. M. Jackson MV is a classic performance of Michael Jackson with spectacular stage effects. The critical backlight levels vary frequently, but the changes are small. BBC News is a news video clip mainly comprised of static scenes.

The performance metric was the percentage of energy savings achieved without adversely impacting the visual perception. We investigated the impacts of each of the three parameters on the algorithm's performance: (1) the impacts of the SSIM index in the range 0.8 to 0.98 when the minimum duration d was set at 10 frames and the differential ratio r was set at 0.5; (2) the impacts of the differential ratio in the range 0.1 to 1 when the SSIM index was set at 0.9 and d was set at 10 frames; and (3) the impacts of setting the minimum duration between 5 and 50 frames when the SSIM index was set at 0.9 and r was set at 0.5. The settings were based on the following observations. HTC Desire allows a backlight change every four frames for 30 frame-per-second videos. Moreover, it was difficult to determine whether a video adopted our technique when the SSIM index was no less than 0.8; and it was difficult to differentiate which of two videos adopted our technique when the SSIM index was no less than 0.9. The difficulty is more evident when the differential ratio is no larger than 0.5.

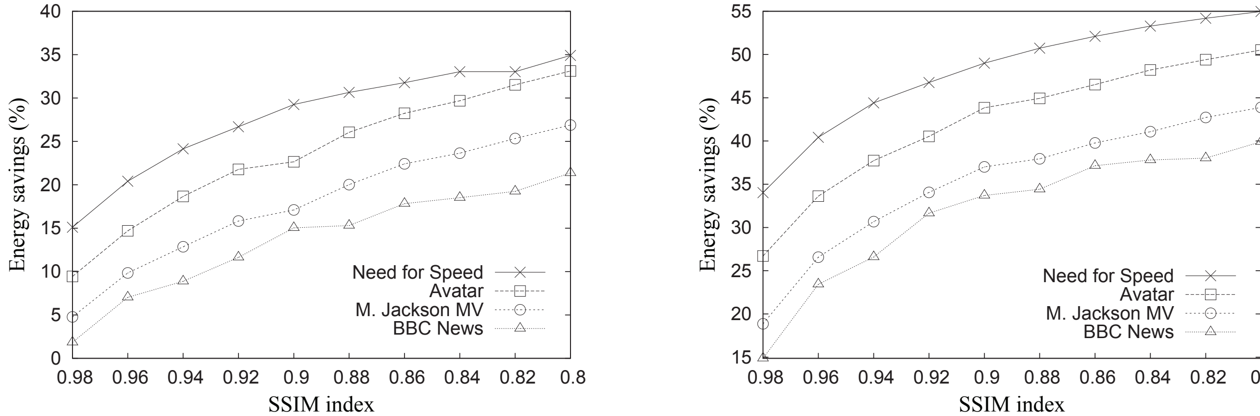
In addition to the above studies, we compared the proposed algorithm, denoted as *OPT*, with a heuristic-based algorithm, denoted as *GOS*, which is based on the concept of groups of scenes utilized by the approach in [23]. GOS starts with a group of d frames, and then adds subsequent frames until the variance of the average luminosity exceeds a threshold. The process is repeated until all frames have been grouped, and the frames in each group are assigned the maximum of their critical backlight levels. Following the settings in [23], the variation threshold was set at 40. We report the results of experiments when the SSIM index was set at 0.9 and the differential ratio r was set at 0.5, while the minimum duration d was set at 10, 30, and 50. GOS was adopted for comparison because the minimum duration between changes in backlight levels can be ensured by setting the minimum group size as d. Finally, we conducted experiments to assess the time required to exhaust the battery. Two identical mobile devices, HTC Desire or Apple iPad, were fully recharged and used to play repeatedly the same set of 42 videos selected at random from YouTube until their battery was exhausted. The experiment was performed in an off-line fashion to ensure that the time measured was not affected by unstable wireless communications.

5.2 Experimental Results

Figure 9 shows the impacts of the SSIM index on the energy savings achieved by OPT. As expected, the energy savings increased as the SSIM index decreased. The reason was that a smaller SSIM index led to lower critical backlight levels, and this in turn implied more energy savings. The results show that, when the SSIM was set at 0.9, OPT can achieve energy savings of 15% to 29% for HTC Desire and 34% to 49% for Apple iPad, depending on the characteristics of videos. A more evident energy reduction was achieved on Apple iPad than on HTC Desire, because the slope of the former's power model increased more quickly with the backlight levels. We also observe that the percentage of energy savings was more evident when a video's critical backlight levels varied significantly, such as in the Need for Speed video. This was because the significant variation meant a large number of low critical backlight levels, which provided opportunities to dim the backlight. Interestingly, a video with a large number of static scenes, such as BBC News, also benefited substantially from the dynamic backlight scaling technique due to the difference between the full and critical backlight levels.

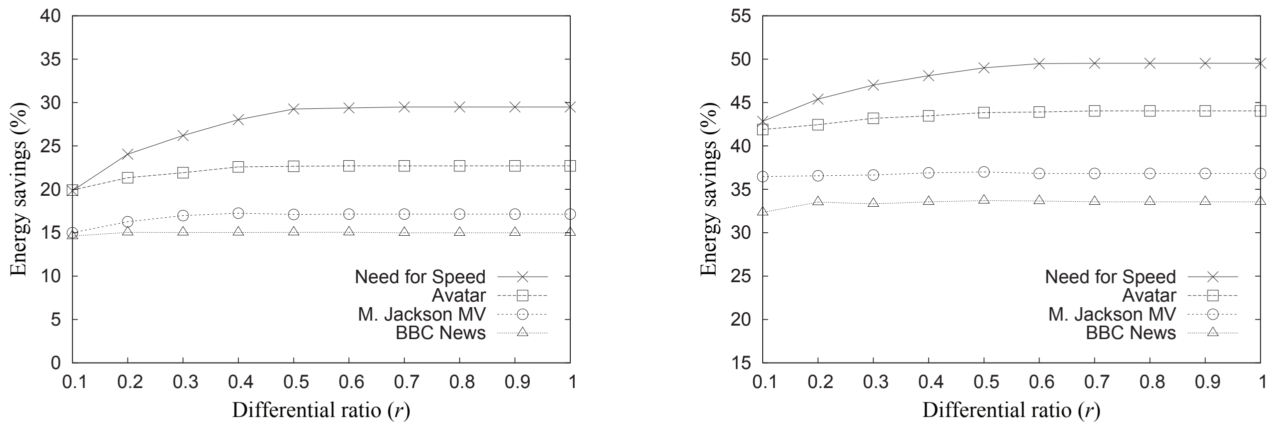
Figure 10 shows the impacts of the differential ratio on the energy savings achieved by OPT. The energy savings increased as the differential ratio increased. The result was as expected because, for the same backlight level, a larger differential ratio allowed a larger backlight

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



(a) HTC Desire (b) Apple iPad

Fig. 9. Impacts of the SSIM index



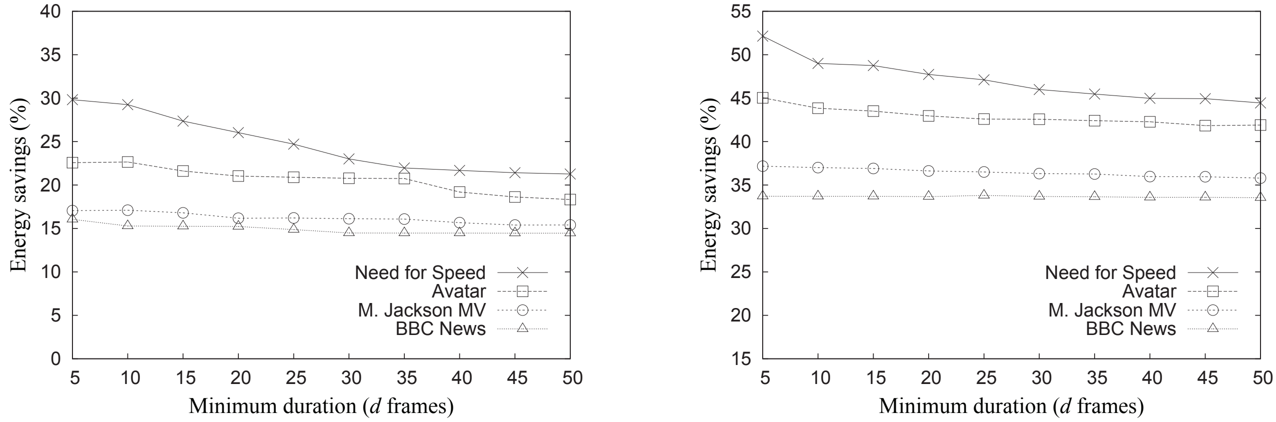
(a) HTC Desire (b) Apple iPad

Fig. 10. Impacts of the differential ratio

change. The characteristic was specially beneficial for videos whose critical backlight levels varied significantly to save energy. This also explains why the differential ratio had a more significant impact on the Need for Speed video than on the other videos. The results show that the impacts of the differential ratio became saturated when $r \geq 0.5$. In addition, we observe that the difference in energy savings between the Need for Speed and Avatar videos was very small when r was set at 0.1. This phenomena occurred because the Need for Speed video comprised a large number of short scenes, and the possible backlight decreasing for the scenes was limited by a small differential ratio.

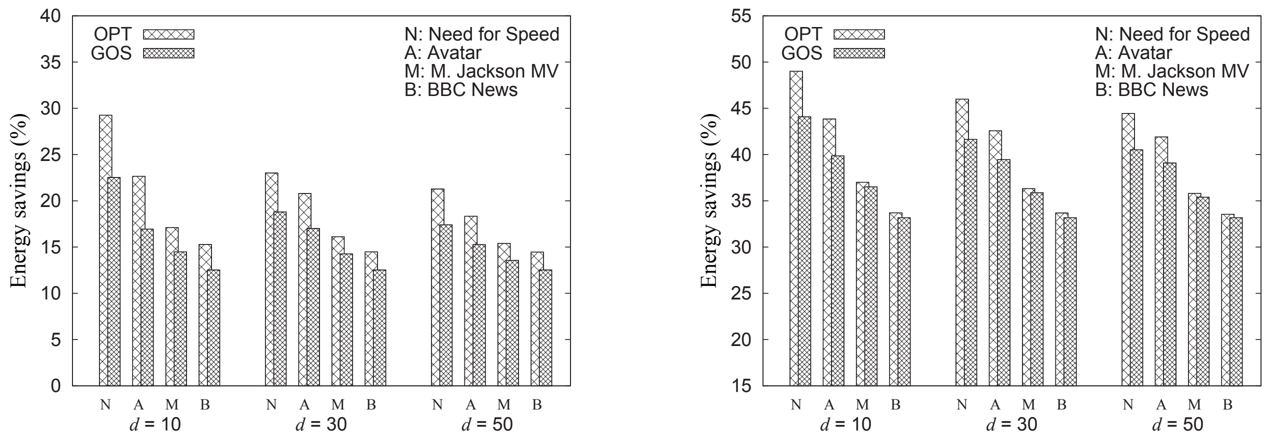
Figure 11 shows the impacts of the minimum duration on the energy savings achieved by OPT. The energy savings generally decreased as the minimum duration increased. This is because the solution space under a larger duration is a subset of that under a smaller duration. We observe that the minimum duration affected the energy savings of the Need for Speed and Avatar videos significantly; however, it did not have a significant impact on M. Jackson MV and BBC News. The results imply that the first two videos contained a large variety of scenes, each comprised of a few frames. In contrast, the other two videos contained longer scenes with similar critical backlight levels. In addition, as in the previous experiments, the more diverse the critical backlight levels, the greater the energy savings. To conclude the impacts of the parameters on energy savings based on the above experiments, the impacts appeared more significantly by varying the SSIM index than by varying the differential ratio or the minimum duration.

Figure 12 shows the energy savings achieved by OPT and GOS for videos with different characteristics. As expected, OPT outperformed GOS in all cases because it has proved optimal in terms of energy savings. For the same video, the performance difference between the two algorithms was generally more evident when the minimum duration was small, since the number of possible solutions increased as the minimum duration decreased. Consequentially, it was harder for a heuristic algorithm, GOS for example, to make a "good" decision. This also explains why, for the same minimum duration, the performance difference was generally more evident when a video's critical backlight levels varied significantly. The results show that, for the Need for Speed video on HTC Desire, OPT reduced the energy consumption 1.3, 1.23, and 1.22 times more than GOS when d was set at 10,



(a) HTC Desire (b) Apple iPad

Fig. 11. Impacts of the minimum duration



(a) HTC Desire (b) Apple iPad

Fig. 12. Energy savings achieved by OPT and GOS for different videos

	HTC Desire	Apple iPad
Display Subsystem	15-29%	34-49%
Whole Device	10-20%	27-40%
Prolonged Time	42 minutes	3 hours + 31 minutes

TABLE 1
Energy savings and prolonged times

30, and 50, respectively. Importantly, we observed that, under GOS, a scene may suffer from sudden backlight changes if it covered two groups and the maximum critical backlight levels of the groups were very different.

Table 1 shows the energy savings achevied and the usage times prolonged by OPT. The results of experiments conducted over a set of videos show that HTC Desire achieved energy savings of 15% to 29% for the display subsystem and 10% to 20% for the whole device, while Apple iPad achieved energy savings of 34% to 49% and 27% to 40%, respectively. For Apple iPad, a tremendous amount of energy was reduced, because it was equipped with a large display subsystem that accounted for the majority of the energy consumption of the whole device. On the other hand, the results of experiments conducted to exhaust the battery show that, for HTC Desire, the smartphone without dynamic backlight scaling ran out of energy after 3 hours and 31 minutes, while the smartphone that utilized the technique ran out after 4 hours and 13 minutes, a difference of 42 minutes. For Apple iPad, the tablet without dynamic backlight scaling was forced to turn down (when the remaining energy was lower than 10 percent) after 6 hours and 6 minutes, while the tablet that utilized the technique was forced to turn down after 9 hours and 37 minutes, a difference of 3 hours and 31 minutes. Both the devices benefited substantially from the dynamic backlight scaling technique.

6 Concluding Remarks

This paper proposes an approach that minimizes the energy consumption incurred by the backlight when users access multimedia streaming on mobile devices. Specifically, the approach exploits backlight scaling and models a fundamental optimization problem with scaling constraints (to limit image distortion, reflect hardware limitation, and consider user perception). To solve the problem, we propose three algorithms, and prove that they are optimal in terms of energy savings when the energy consumption increases strictly with the backlight levels. To validate the practicability of our approach, based on the algorithms, we have deployed a cloud-based energy-saving service, called the dynamic backlight scaling service, on CHT hicloud [3]. We have also implemented a mobile application program that enables Android smartphones [1] and Apple tablets [2] to access the energy-saving service. With the program installed, HTC Desire [6] and Apple iPad [2] could achieve energy savings of 15-29% and 34-49% respectively (in items of the energy consumption of the display subsystems), when browsing videos on YouTube [8], while users were not conscious of the dynamic backlight scaling technique. The efficacy of the proposed approach is more evident for large-screen mobile devices or when a video contains a large variety of scenes. Moreover, we have released the mobile application program in the Hami Apps [5] to seek feedback on the performance and to identify issues that require further investigation.

References

[1] Android. `http://www.android.com`.

[2] Apple Inc. `http://www.apple.com`.

[3] CHT hicloud CaaS. `http://hicloud.hinet.net`.

[4] Facebook. `http://www.facebook.com`.

[5] Hami Apps. `http://halink.emome.net/index_en_US.html`.

[6] HTC Corp. `http://www.htc.com`.

[7] Monsoon Solutions, Inc. `http://www.msoon.com`.

[8] YouTube. `http://www.youtube.com`.

[9] T. Acharya and A. K. Ray. *Image Processing: Principles and Applications*. 2005.

[10] A. Bartolini, M. Ruggiero, and L. Benini. HVS-DBS: Human Visual System-Aware Dynamic Luminance Backlight Scaling for Video Streaming Applications. In *Proc. of the ACM EMSOFT*, pages 21–28, 2009.

[11] N. Chang, I. Choi, and H. Shim. DLS: Dynamic Backlight Luminance Scaling of Liquid Crystal Display. *IEEE Trans. on VLSI Sys.*, 12(8):837–846, 2004.

[12] L. Cheng, S. Mohapatra, M. E. Zarki, N. Dutt, and N. Venkatasubramanian. Quality-Based Backlight Optimization for Video Playback on Handheld Devices. *Advances in Multimedia*, 2007.

[13] W.-C. Cheng and C.-F. Chao. Minimization for LED-backlit TFT-LCDs. In *Proc. of the ACM/IEEE DAC*, pages 608–611, 2006.

[14] W.-C. Cheng and M. Pedram. Power Minimization in a Backlit TFT-LCD Display by Concurrent Brightness and Contrast Scaling. *IEEE Trans. on Consumer Electronics*, 50(1):25–32, 2004.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, pages 162–164, 289–292. The MIT Press, 3 edition, 2009.

[16] V. Devadas and H. Aydin. On the Interplay of Voltage/Frequency Scaling and Device Power Management for Frame-Based Real-Time Embedded Applications. *IEEE Trans. on Computers*, 61(1):31–44, 2012.

[17] C.-W. Huang and S.-L. Tsao. Minimizing Energy Consumption of Embedded Systems via Optimal Code Layout. *IEEE Trans. on Computers*, 61(8):1127–1139, 2012.

[18] A. Iranli, H. Fatemi, and M. Pedram. HEBS: Histogram Equalization for Backlight Scaling. In *Proc. of the ACM/IEEE DATE*, pages 346–351, 2005.

[19] A. Iranli, W. Lee, and M. Pedram. Backlight Dimming in Power-Aware Mobile Displays. In *Proc. of the ACM/IEEE DAC*, pages 604–607, 2006.

[20] A. Iranli, W. Lee, and M. Pedram. HVS-Aware Dynamic Backlight Scaling in TFT-LCDs. *IEEE Trans. on VLSI Sys.*, 14(10):1103–1116, 2006.

[21] A. Iranli and M. Pedram. DTM: Dynamic Tone Mapping for Backlight Scaling. In *Proc. of the ACM/IEEE DAC*, pages 612–616, 2005.

[22] D. Niu, Y. Chen, and Y. Xie. Low-power Dual-element Memristor Based Memory Design. In *Proc. of the ACM/IEEE ISLPED*, pages 25–30, 2010.

[23] S. Pasricha, M. Luthra, S. Mohapatra, N. Dutt, and N. Venkatasubramanian. Dynamic Backlight Adaptation for Low-Power Handheld Devices. *IEEE Design & Test of Computers*, 21(5):398–405, 2004.

[24] F. Paterna, A. Acquaviva, A. Caprara, F. Papariello, G. Desoli, and L. Benini. Variability-Aware Task Allocation for Energy-Efficient Quality of Service Provisioning in Embedded Streaming Multimedia Applications. *IEEE Trans. on Computers*, 61(7):939–953, 2012.

[25] N. Petti and Y.-H. Lu. A Homogeneous Architecture for Power Policy Integration in Operating Systems. *IEEE Trans. on Computers*, 58(7):945–955, 2009.

[26] M. Ruggiero, A. Bartolini, and L. Benini. DBS4video: Dynamic Luminance Backlight Scaling based on Multi-Histogram Frame Characterization for Video Streaming Application. In *Proc. of the ACM EMSOFT*, pages 109–118, 2008.

[27] A. Shye, B. Scholbrock, and G. Memik. Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures. In *Proc. of the ACM/IEEE MICRO*, pages 168–178, 2009.

[28] A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda. Characterizing and Modeling User Activity on Smartphones: Summary. In *Proc. of the ACM SIGMETRICS*, pages 375–376, 2010.

[29] P.-S. Tsai, C.-K. Liang, T.-H. Huang, and H. H. Chen. Image Enhancement for Backlight-Scaled TFT-LCD Displays. *IEEE Trans. on Circuits and Systems for Video Technology*, 19(4):574–583, 2009.

[30] E. H. Weber. *De Pulsu, Resorpitione, Auditu et Tactu: Annotationes Anatomicae et Physiologicae*. 1834.

[31] Y.-J. Yu, P.-C. Hsiu, and A.-C. Pang. Energy-Efficient Video Multicast in 4G Wireless Systems. *IEEE Trans. on Mobile Computing*, 11(10), 2012.

[32] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. on Image Processing*, 13(4):600–612, 2004.

Chun-Han Lin received the BS and MS degrees in computer information science from National Chiao Tung University in 2000 and 2002, respectively, and PhD degree in computer science from National Tsing Hua University in 2010. He is currently a postdoctoral fellow in the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taiwan. His research interests include networked embedded systems and networked sensor systems. He is a member of the IEEE.

Pi-Cheng Hsiu received the BS degree in computer information science from National Chiao Tung University in 2002 and the MS and PhD degrees in computer science and information engineering from National Taiwan University in 2004 and 2009, respectively. He is currently an assistant research fellow in the Research Center for Information Technology Innovation (CITI) and also a joint assistant research fellow in the Institute of Information Science (IIS), Academia Sinica, Taiwan. He was a visiting scholar in the Department of Computer Science at the University of Illinois at Urbana-Champaign in 2007. His research interests include networked embedded systems, real-time systems, and wireless mobile networks. He is a member of the IEEE and the IEEE Computer Society.

Cheng-Kang Hsieh received the BS degree in computer science and information engineering from National Taiwan University in 2009. In 2011, he was a research assistant in the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taiwan. He is currently a PhD student in the Department of Computer Science at the University of California, Los Angeles, USA. His research interests include networked embedded systems and storage systems.