

Smart Lighting Control Using Oblivious Mobile Sensors

Areg Karapetyan

Department of Computer Science
Masdar Institute, Khalifa University
Abu Dhabi, UAE
areg.karapetyan@ku.ac.ae

Khaled Elbassioni, Majid Khonji

Department of Computer Science
Masdar Institute, Khalifa University
Abu Dhabi, UAE
{khaled.elbassioni,majid.khonji}@ku.ac.ae

Sid Chi-Kin Chau

Research School of Computer Science
Australian National University
Canberra, Australia
sid.chau@anu.edu.au

Emad Dababseh

Abu Dhabi, UAE
emad.dababseh@gmail.com

ABSTRACT

The interplay of smart light bulbs (equipped with wireless controllable LEDs) and mobile sensors (embedded in wearable devices, such as smart watches and spectacles) enables a wide range of interactive lighting applications. One notable example is a *smart lighting control system* that provides automated illuminance management by wearable sensors close to end-users. In this paper, an energy-efficient smart lighting control system is developed using mobile light sensors for measuring local illuminance and assisting smart light bulbs to coordinate the brightness adjustments, while meeting users' heterogeneous lighting preferences. A pivotal challenge in these systems is attributed to the presence of *oblivious mobile sensors* hampered by the uncertainties in their relative locations to light bulbs, unknown indoor environment and time-varying background light sources. To cope with these hindrances, we devise an effective model-agnostic control algorithm inducing continuous adaptive coordination of oblivious mobile sensors without complete knowledge of dynamic operational environment and the associated parameters. The proposed algorithm is corroborated extensively under diverse settings and scenarios in a proof-of-concept smart lighting testbed featuring programmable light bulbs and smartphones, deployed as light sensing units. Lastly, we discuss some practical limitations of the proposed control approach, along with possible solutions, and conclude by outlining promising directions for future work.

CCS CONCEPTS

• **Human-centered computing** → **Ambient intelligence**; *Ubiquitous and mobile computing*; • **Theory of computation** → **Stochastic control and optimization**; *Online learning algorithms*; • **Computing methodologies** → **Optimization algorithms**; • **Computer systems organization** → **Sensors and actuators**; • **Mathematics of computing** → *Simulated annealing*;

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.

BuildSys '18, November 7–8, 2018, Shenzhen, China

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5951-1/18/11...\$15.00

<https://doi.org/10.1145/3276774.3276788>

KEYWORDS

Smart Lighting Control, Oblivious Mobile Sensors, Internet-of-Things, Illuminance Control Algorithm, Wearable Computing.

1 INTRODUCTION

Modern lighting systems are undergoing a dramatic transformation from traditional incandescent lights to energy-efficient LED technology, equipped with advanced power electronics to control brightness and multi-color LEDs to simulate a dynamic range of colors. These controllable LEDs are also integrated with system-on-a-chip to provide local computation and wireless connectivity via popular wireless network protocols (e.g., WiFi, Bluetooth, ZigBee), thus creating a new class of Internet-of-things (IoT) devices, called *smart light bulbs* (e.g., Philips Hue, LIFX).

Smart light bulbs can enable innovative lighting applications, for example, personalizable context-driven lighting control that can synchronize with the picture and music on TV. Notably, smart bulbs can effectively simulate natural light, providing more pleasant illuminance for indoor environments. In future, OLED technology can empower smart bulbs to have a more vibrant color range with flexible structures and shapes, which can be integrated with furniture, decorations and sculptures for artistic designs and infotainment displays. Therefore, smart bulbs with their increasing affordability are becoming indispensable in smart buildings.

Like many other IoT devices, smart bulbs aim to bolster the intelligent management and control abilities of building facilities and infrastructures, such as in smart buildings for improving user comfort and energy efficiency. Typically, an IoT device requires a close integration of sensing and control technologies, which can be deployed in ad-hoc dynamic environments. For the application of illuminance management, it is critical to integrate with diverse sources of sensing data. Particularly, there are advances in mobile sensing technology that create smaller low-powered sensors embedded in wearable devices, such as smart watches, wristbands, clothes and spectacles. Thus, the combination of smart bulbs and mobile sensors can enable a wide range of novel interactive lighting applications.

However, conventional user-operated control systems for lighting are ineffective to cope with dynamic environments and users' heterogeneous preferences [12, 17]. Hence, IoT-based smart lighting systems are proposed to provide more resilient control by adapting

to the time-varying environment as well as catering for heterogeneous contexts and users' preferences in real time. Essentially, these systems rely on the interplay of wireless controllable smart bulbs and mobile sensors to continuously adjust the lighting configurations (e.g., brightness levels, colors, orientations) in response to the environmental factors and users' preferences. Meanwhile, mobile sensors (possibly embedded in wearable devices like smart watches and spectacles) can be utilized to measure the local illuminance close to end users, and broadcast the measurement data to nearby lighting system controllers. Consequently, smart bulbs, upon receiving the measurement data, will be able to coordinate the brightness and color adjustments automatically, while meeting users' heterogeneous illuminance preferences. For example, consider the scenario illustrated in Fig. 1, where users are carrying mobile sensors in smart wearable devices, which periodically broadcast local illuminance measurement data. Listening to this data, the smart bulbs should be able to figure out the appropriate local illuminance configurations, in spite of the dynamic user movements and unknown background light sources.

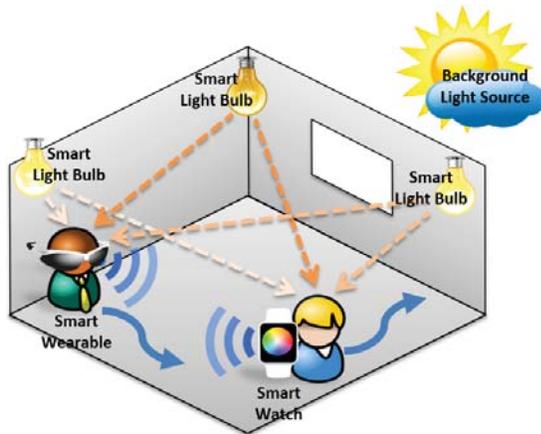


Figure 1: An illustration of smart lighting control system with mobile sensors in smart wearable devices broadcasting local illuminance measurement data, and smart light bulbs coordinating to adjust the brightness levels and colors.

IoT-based smart lighting systems have certain advantages in many scenarios that are difficult to achieve with other, potentially pervasive, methods. For instance, one alternative approach is to use computer vision techniques employing low-cost cameras to detect occupancies in a building. However, camera placement might not be a trivial task especially in dynamic environments that cause occlusion from camera line of sight (e.g., in a warehouse). Also, differentiating occupants from other moving obstacles (e.g., robotic vacuum cleaner, or warehouse robots) could be difficult to achieve using motion detectors, or inaccurate when using computer vision techniques in those environments.

To enable intelligent lighting control, we develop a prototype system with mobile light sensors for measuring local illuminance and assisting smart bulbs to dynamically coordinate the brightness adjustments, subject to users' heterogeneous illuminance preferences. In particular, we consider the presence of *oblivious mobile*

sensors. These sensors are hindered by the uncertainties in their relative locations to light bulbs, the presence of dynamic obstacles in surrounding environment, and time-varying background light sources. Specifically, the major challenges of smart lighting control systems with oblivious mobile sensors are highlighted below.

- **Lack of Localization and Dynamic Environment:** Given complete knowledge of the locations of sensors and light bulbs, one would be able to accurately model the illuminance effects of light bulbs assuming a static surrounding environment. In practice, however, there are various sources of background light (e.g., the sun, outdoor light sources), and dynamic obstacles in the environment. Moreover, portability of wearable sensors limit the availability of their location information as attaining accurate localization of mobile devices for indoor environments is hard. Thus, the perceived illuminance by users is a complicated time-varying function of the surrounding environment and their positions.
- **Ad-hoc Management:** Smart buildings and homes are being frequently retrofitted with new appliances and devices. Only simple set-ups are invoked for these newly deployed components. Thus, smart bulbs and wearable sensors are often required to cope with a variable number of appliances in ad-hoc situations.
- **Computational Complexity:** IoT devices are highly portable with simple electronics and circuits. Smart bulbs and wearable sensors do not have sophisticated processing power. Hence, it is desirable to have computationally savvy and easily implementable control algorithms without relying on complicated learning and tuning processes of the parameters of the environment and involved coordination protocols among the devices.

Taken together, these challenges make the design of effective smart lighting control systems a highly non-trivial problem.

In general, smart lighting control belongs to the class of *oblivious multi-sensor control problems*, which aim to provide effective control without complete knowledge of the dynamic uncertain environment, subject to certain quality-of-service control constraints. There are a wide range of examples of oblivious multi-sensor control problems, such as *environmental sensing and control for air quality, sound quality, temperature and indoor climate* [2]. In this paper, we shall confine our attention particularly on the lighting application, which also shares certain similarities with the aforementioned ones.

To cope with the uncertainties inflicted by oblivious mobile sensors, we devise an effective control algorithm devoid of any assumptions or a priori information on erratic spatio-temporal environmental parameters. The basic idea is based on the notion of learning a dynamic environment by random sampling with only a small number of measurements. Unlike typical sampling-based approaches (e.g., those covered in [4, 7, 9, 10]), which will likely incur a sizable sampling space and a slow convergence rate, the proposed control algorithm rapidly converges to near optimal solutions, by reducing the general problem to a sub-problem with a simpler and well-defined sampling domain. To demonstrate the empirical effectiveness of the featured approach in practice, it is implemented and applied to a proof-of-concept smart lighting system testbed (using

LIFX programmable light bulbs and smartphones) deployed in a real-world indoor environment.

Outline of the Paper: In Sec. 2, we first present the background and related work. Sec. 3 provides the formal definition of the smart lighting control problem with oblivious sensors. Then, an effective control algorithm is devised In Sec. 4 . In Sec. 5, we describe the proof-of-concept smart lighting system testbed implementation, and in Sec. 6, we present the detailed results of experimental evaluations. Finally, Sec. 7 discusses some open practical issues of the featured smart lighting control system.

2 RELATED WORK

Smart lighting control has sparked considerable interest in recent years. With the promising capabilities and enhanced functionalities for energy-efficiency and ambient control, intelligent lighting systems present a number of benefits as well as challenges. In particular, to properly realize the benefits of these systems, diverse factors and practical constrains should be taken into account, such as daylight harvesting, cross-illumination effects and external background light sources as well as users' lighting preferences and contextual requirements.

However, such a detailed model of lighting control incorporating all these factors is often impractical for real-life applications, since it requires extensive computations and sophisticated measurements. Therefore, most of the extant literature on lighting control systems simplified the model by omitting some of the constrains and factors, thereby substantially reducing complexity of the underlying optimization problem. Moreover, heuristic algorithms are often utilized for solving the relaxed problem with sub-optimal solutions.

Along these lines, in [15, 16], a centralized wireless networked lighting system that maximizes energy efficiency and user satisfaction is introduced, based on a linear programming formulation. However, the employed model is highly static and largely limited to the availability of apriori information on the environmental parameters. Another utility-based lighting control strategy is proposed in [13] that optimizes the trade-off between meeting user comfort and minimizing energy usage. This model relies on a simplifying assumption that only a small number of users are affected by a single luminary. A heuristic strategy is utilized for solving the suggested optimization model.

In a recent work [8], a decentralized algorithm is introduced that balances the lighting while minimizing the energy consumption. The authors investigated the impact of intercommunication between light bulbs on the performance of the algorithm. The study in [17] utilized wearable sensors to propose an infrastructure for context-aware lighting, with the aim to improve user comfort, system performance, and energy consumption. In [11], a sensor system is developed that can monitor and control power usage of appliances (including lights) without supervised training. Based on multi-sensor fusion and unsupervised machine learning algorithms, the featured system can classify the appliance events of interest and autonomously associate measured power usage with the respective appliances. However, this study does not directly apply sensor data to lighting control. Finally, several online algorithms for controlling building facilities, which can also applied to lighting control, have been investigated in prior work [1, 5].

3 MODEL FORMULATION OF SMART LIGHTING CONTROL

In this section, we formulate the model of smart lighting control system under study. We remark that this model is sufficiently generic for other applications of environmental sensing and actuation systems (e.g., for air quality, sound quality, temperature etc.).

Formally, we consider n sensors (indexed by the set $\mathcal{N} \triangleq \{1, \dots, n\}$) and m light bulbs (indexed by the set $\mathcal{M} \triangleq \{1, \dots, m\}$) procured by a controller unit over a decision horizon $\mathcal{T} \triangleq \{1, \dots, T\}$ discretized into T equal periods. Let x_i^t be the brightness level¹, which is parameterized by the input wattage, of the i -th light bulb at time $t \in \mathcal{T}$. We will simply write x_i , when referring to a particular time-slot. Let $f_{i,j}^t(x_i^t)$ be an abstract function mapping the input power of i -th light bulb to the recorded illuminance measurement at the j -th sensor at time t .

Note that the functions $f_{i,j}^t(\cdot)$ depend on the locations of sensors and light bulbs, as well as background light sources and other environmental dynamic factors. Furthermore, in general, a sensor can detect only the aggregate illuminance from the light bulbs and sources present and not from a specific luminary (unless otherwise all the other sources are intentionally switched off). Thus, these mapping functions are time-varying and unknown, which in a sense, represents the complex dependence of indoor environment. Accordingly, we model $f_{i,j}^t(\cdot)$ as abstract functions given via an *evaluation oracle* (i.e., function values are accessible at queried points only, while the exact function is not known). On one hand, this choice of mapping functions notably amplifies the flexibility and robustness of the developed model, thereby alleviating the major concerns present in the surveyed literature. On the other hand, under such stringent formulation, the toolbox of applicable solution techniques is largely pruned.

The proposed control model for smart lighting can be divided into the following two consecutive stages.

- (1) *Bootstrapping:* Under the paucity of available apriori knowledge on the environment (e.g., at the start-up stage or after abrupt changes detected in the environment), the sensors and light bulbs need to infer the optimal initial configurations. This is attained by solving problem 2 defined in the proceeding subsection.
- (2) *Continual Adjustment:* To assure continuous adaptive control after the bootstrapping stage, the sensors detect small changes in the environment in real-time and the the light bulbs perform minor adaption accordingly.

In this model, it is assumed that the sensors can communicate with light bulbs using wireless communications (e.g., ZigBee, WiFi) through a simple broadcast channel, and each light bulb can listen to all the sensors. Note that time synchronization can be handled by a proper multi-access protocol (e.g., exponentially back-off CSMA), such that the sensors can transmit the readings without collision of data packets.

¹This can also be other properties, such as color or temperature of color.

3.1 Bootstrapping

In the bootstrapping phase the sensors and light bulbs are unable to rely on any apriori knowledge of the environment due to substantial uncertainties present therein. Thus, to determine the optimal brightness configurations $(x_i^t)_{i \in \mathcal{M}}$ of light bulbs at time $t \in \mathcal{T}$, this stage solves the *multi-sensor lighting control problem* (LCP) defined by the following oracle optimization problem

$$\begin{aligned} \text{(LCP)} \quad & \min_{(x_i^t)_{i \in \mathcal{M}}} \sum_{i=1}^m x_i^t \\ \text{subject to} \quad & \sum_{i=1}^m f_{i,j}^t(x_i^t) \geq c_j^t, \quad \forall j \in \mathcal{N} \quad (1) \\ & x_i^t \geq 0, \quad \forall i \in \mathcal{M}, \quad (2) \end{aligned}$$

where $(c_j^t)_{j \in \mathcal{N}}$ are users' heterogeneous illuminance preferences and Const. (1) is specified only by a *membership oracle* (that is, a procedure which, given any point, determines whether or not it belongs to the feasible set). Without loss of generality, assume $c_j^t = 1$ for $\forall j \in \mathcal{N}$ as Const. (1) can be normalized to have 1 in the right hand side when dividing its both sides by c_j^t . With a slight abuse of notation, in what follows we shall sometimes refer to a vector of variables without subscript (e.g., $x^t \triangleq (x_i^t)_{i \in \mathcal{M}}$) and denote the aggregate illuminance at the j -th sensor by $f_j^t(x^t) \triangleq \sum_{i=1}^m f_{i,j}^t(x_i^t)$.

As such, LCP seeks to meet users' illuminance requirements at the lowest possible net energy consumption of light bulbs. However, the standard optimization techniques are not amenable to LCP, as its constraint matrix is unknown (because of oblivious sensors). The putative method, in fact, given $f_j^t(\cdot)$ (as only the aggregate illuminance can be measured) should learn each function $f_{i,j}^t(\cdot)$ simultaneously while computing the control decisions.

One viable solution, whose conceptual illustration appears in Fig. 2, is based on learning and random sampling by a subset of measurements. The basic idea is to generate brightness levels of light bulbs $\{x', x'', \dots\}$ according to a certain rule. Then, measure the corresponding aggregate illuminance $\left\{ (f_j^t(x'))_{j \in \mathcal{N}}, (f_j^t(x''))_{j \in \mathcal{N}}, \dots \right\}$ at each sensor and broadcast the readings to the controller unit. Next, the light bulbs adjust the brightness levels based on the inferred information. This pipeline is applied iteratively until the illuminance levels recorded at the sensors converge to the desirable values specified by users.

However, when applied to LCP directly, typical sampling-based methods (e.g., Simulated Annealing algorithm [7]) will likely induce a large sampling space and a slow convergence rate. Against this backdrop, in the sequel we present an efficient reduction that transforms LCP to an equivalent problem with a simpler feasibility region, allowing more efficient and rapid sampling. A peculiar advantage of this reduction, over tackling the original problem directly, is that the feasibility region of the resultant problem is a regular polytope. This result is interesting by itself as it can potentially conduce to devising more efficient algorithms for oracle optimization problems.

Specifically, first, LCP is decomposed into several sub-problems through the following transformation.

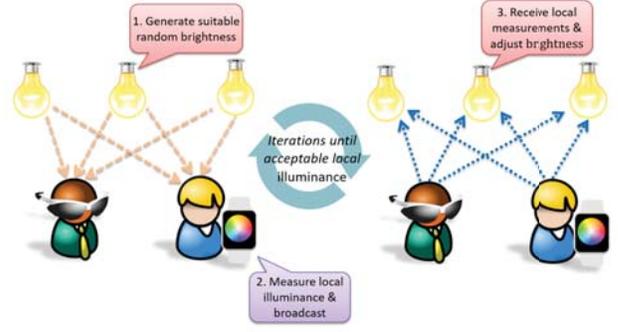


Figure 2: A high-level illustration of the bootstrapping control.

Let $S(B) \triangleq \{x \mid \sum_{i=1}^m x_i = B, x_i \geq 0\}$ be a feasible set with a total brightness level equal to the value B . Then, the sub-problem defined for a given B takes the form

$$\begin{aligned} \text{(LCP2}[B]) \quad & \max_{x^t, \lambda} \lambda \\ \text{subject to} \quad & \sum_{i=1}^m f_{i,j}^t(x_i^t) \geq \lambda, \quad \text{for } \forall j \in \mathcal{N} \quad (3) \\ & x^t \in S(B). \quad (4) \end{aligned}$$

Let $\lambda^*[B]$ be the optimal objective value of LCP2[B]. Notice that LCP is equivalent to finding the smallest B such that $\lambda^*[B] = 1$, signifying that one can solve LCP through binary search on the appropriate value of B .

Next, each sub-problem is further reduced to an easier instance, whose feasibility region is limited to the simplex $S(B)$. More concretely, LCP2[B] can be reduced to an equivalent *max-min resource sharing problem* of the form

$$\lambda^*[B] = \min\{\Lambda(p) \mid p \in \mathbb{R}_+^n, \sum_{j=1}^n p_j = 1\}, \quad (5)$$

where $\Lambda(p) \triangleq \max\{\sum_{j=1}^n p_j \sum_{i=1}^m f_{i,j}^t(x_i^t) \mid x^t \in S(B)\}$. Hence, if one can solve problem (5) efficiently, then can one also solve LCP.

It should be noted that the above reduction presumes $f_j^t(\cdot)$ to be continuous concave functions. To the best of our knowledge, there doesn't exist any known polynomial-time algorithm for LCP in the literature devoid of assumptions concerning these mapping functions. Moreover, this assumption is fairly general, encompassing a wide range of functions.

3.2 Continual Adjustment

In practice, small changes in the environment occur more often than abrupt ones, for example, when users make stationary movements, or the background light sources (e.g., the sun) gradually change their intensity. In such a case, invoking the above bootstrapping control with random sampling to learn the new environment would be futile. Instead, a heuristic tuning process can be utilized to adjust the brightness slowly according to small environmental changes.

Particularly, the continual adjustment stage tunes the parameters in a greedy manner. Let Δf_j^t be the detected change in the illuminance measurement of the j -th sensor at time instant $t \in \mathcal{T}$. This may incur the violation of Const. (1) yielding

$$\Delta f_j^t + \sum_{i=1}^m f_{i,j}^t(x_i^t) < 1.$$

To restore feasibility of the constraint, one can perturb the brightness level of each luminary by an amount Δx , such that

$$\Delta f_j^t + \sum_{i=1}^m f_{i,j}^t(x_i^t + \Delta x) \geq 1,$$

where Δx commensurates with the value of the maximum component of $(\Delta f_j)_{j \in \mathcal{N}}$.

4 ILLUMINANCE CONTROL ALGORITHMS

In this section, we describe an effective control algorithm for oblivious mobile sensors deployed within the envisioned intelligent lighting system, which has been implemented in a proof-of-concept testbed to be described in the subsequent section.

4.1 Bootstrapping Control Algorithm

Recall that the bootstrapping stage aims at tackling LCP as to determine the initial optimal brightness configurations of light bulbs that satisfy users' heterogeneous lighting preferences. As previously noted in Section 3, one computationally efficient approach would be to solve problem 5 iteratively for each value of B returned by binary search on the interval $[0, \bar{B}]$, where \bar{B} is the expected upper bound on the objective value of LCP. Below, we present an effective algorithm, explained in Algs. 1 and 2, capable of producing near optimal approximately feasible solutions to LCP.

The devised control approach starts with a standard binary search algorithm, detailed in Alg. 1, on the values of B . The latter as a subroutine invokes an adapted variant of the Lagrangian decomposition algorithm introduced in [6]. Execution of this subroutine begins with a trivially feasible solution, and then iterates until the relative error μ between the current and previous objective values, defined in step 9, drops below the desired accuracy characterized by the input parameter η . In a typical iteration, a set of weights $\{p_j\}_{j \in \mathcal{N}}$ corresponding to the set of inequalities in (3) is computed (step 4). These weights are chosen carefully such that the weighted average of the left-hand sides of these inequalities (precisely, $\sum_{j=1}^n p_j \sum_{i=1}^m f_{i,j}^t(x_i^t)$) closely approximates $\min_{j \in \mathcal{N}} \sum_{i=1}^m f_{i,j}^t(x_i^t)$. The rationale behind this is that maximizing the weighted average of these mapping functions is essentially equivalent to maximizing their minimum, which, in turn, is analogous to minimizing the violations of users' lighting preferences.

Based on these weights, the algorithm then in step 8 selects an \hat{x} , among the candidates sampled uniformly from $\mathcal{S}(B)$, that approximately maximizes the mentioned objective. Next, it computes the error μ and the update step size τ (steps 9 and 10, respectively), then updates the current solution x as described in step 11. This yields a new feasible solution, thereby concluding the iteration. Note that the step size τ can be altered depending on the application.

An integral part of the above pipeline is the sampling phase in step 7, which entails generating z points from the m -dimensional

Algorithm 1 BinarySearch $[\epsilon, \eta, \bar{B}, (f_j^t)_{j \in \mathcal{N}}]$

```

1:  $B_l \leftarrow 0; B_u \leftarrow \bar{B}$ 
2:  $\hat{x}_i \leftarrow \frac{B}{m} \quad \forall i \in \mathcal{M}$ 
3: repeat
4:    $B \leftarrow \left(\frac{B_l + B_u}{2}\right)$ 
5:    $(\lambda, (x_i)_{i \in \mathcal{M}}) \leftarrow \text{BootStrapCtrl}[\eta, B, (f_j^t)_{j \in \mathcal{N}}, \hat{x}, 2m]$ 
6:    $\hat{x} \leftarrow x$ 
7:   if  $\lambda < 1$  then
8:      $B_l \leftarrow B$ 
9:   else
10:     $B_u \leftarrow B$ 
11:  end if
12: until  $|\lambda - 1| < \epsilon$ 
13: return  $(x_i)_{i \in \mathcal{M}}$ 

```

Algorithm 2 BootStrapCtrl $[\eta, B, (f_j^t)_{j \in \mathcal{N}}, x, z]$

```

1:  $\mu \leftarrow \frac{\eta}{6} + 1$ 
2: while  $\mu > \frac{\eta}{6}$  do
3:   Compute  $\theta$  such that
4:     
$$\frac{\epsilon \theta}{6m} \sum_{j=1}^m \frac{1}{f_j^t(x^t) - \theta} = 1.$$

5:    $p_j \leftarrow \frac{1}{m} \frac{\theta}{f_j^t(x^t) - \theta} \quad \forall j \in \mathcal{N}$ 
6:    $\Lambda((\hat{x}_i)_{i \in \mathcal{M}}) \triangleq \sum_{j=1}^n p_j f_j^t(\hat{x})$ 
7:    $\mathcal{S}(B) \triangleq \{(x_i)_{i \in \mathcal{M}} \mid \sum_{i=1}^m x_i = B, x_i \geq 0\}$ 
8:   Generate a uniformly random set
9:     
$$\phi \triangleq \{(x^t)_i)_{i \in \mathcal{M}}, (x'')_i)_{i \in \mathcal{M}}, \dots\}$$

10:   with cardinality of  $z$ , where each item  $\in \mathcal{S}(B)$ .
11:   Find
12:     
$$(\hat{x}_i)_{i \in \mathcal{M}} \triangleq \operatorname{argmax}_{y \in \phi} \left\{ \Lambda(y) \mid y \in \mathcal{S}(B) \right\}.$$

13:   
$$\mu \leftarrow \frac{\sum_{j=1}^n p_j f_j^t(\hat{x}) - \sum_{j=1}^n p_j f_j^t(x)}{\sum_{j=1}^n p_j f_j^t(\hat{x}) + \sum_{j=1}^n p_j f_j^t(x)}$$

14:   
$$\tau \leftarrow \frac{\theta \mu}{2n(\sum_{j=1}^n p_j f_j^t(\hat{x}) + \sum_{j=1}^n p_j f_j^t(x))}$$

15:    $x_i \leftarrow (1 - \tau)x_i + \tau \hat{x}_i \quad \forall i \in \mathcal{M}$ 
16: end while
17:  $\lambda \leftarrow \min_{j \in \mathcal{N}} f_j^t(x)$ 
18: return  $(\lambda, (x_i)_{i \in \mathcal{M}})$ 

```

simplex $\mathcal{S}(B) \triangleq \{x \mid \sum_{i=1}^m x_i = B, x_i \geq 0\}$ uniformly at random, where z is an input parameter of Alg. 2. In a sense, this is equivalent to sampling from a Dirichlet distribution and then normalizing. There are several approaches to address this step, as described in [14], yet the most computationally elegant solution is to generate

z independent points E_1, E_2, \dots, E_z from an exponential distribution² and then divide each sample by their sum (i.e., $\sum_{i=1}^z E_i$). The resultant vector is uniformly distributed over the simplex. Unlike the alternative approaches mentioned in [14], the outlined procedure requires only $O(zm)$ running time. As in the case with τ , the parameter z can also be adjusted according to the execution settings.

An imperative advantage of Alg. 2, as compared to the other sampling-based methods mentioned previously, is its scalability. While Simulated Annealing algorithm requires $O^*(m^{4.5})$ operations³ to terminate [7], the empirical evidence suggests that, under judicious choice of input parameters, Alg. 2 runs notably faster. Observe that, when setting $z = O(m)$, each while loop iteration of Alg. 2 runs in time $O(nm^2)$. Nevertheless, we remark that the overall number of iterations specified in [6] for Lagrangian decomposition algorithm does not hold for Alg. 2, which calls for a more elaborate analysis.

4.2 Continual Adjustment Algorithm

We next describe the continual adjustment algorithm, explained in Alg. 3. Given the illuminance measurement data at time t , namely the aggregate illuminance before and after environmental changes (i.e., $(f_j^{t-1}(x^{t-1}))_{j \in \mathcal{N}}$ and $(f_j^t(x^{t-1}))_{j \in \mathcal{N}}$, respectively), Alg. 3 measures the discrepancy

$$\Delta f_j^t \leftarrow \min \left\{ f_j^t(x^{t-1}) - f_j^{t-1}(x^{t-1}), 0 \right\},$$

and discards positive Δf_j^t (i.e., when illuminance constraint is satisfied).

Then, it updates the brightness levels based on a simple linear heuristic as follows

$$x_i^t \leftarrow x_i^{t-1} + \alpha \cdot \max_j |\Delta f_j^t|,$$

where α is a pre-defined parameter indicating the update step size. If the illuminance constraints remain still unsatisfied (i.e., $f_j^t(x^t) < 1$ for some $j \in \mathcal{N}$), then the update rule is applied iteratively. Note that the running time of Alg. 3 is linear in n and m , as $\max_{j \in \mathcal{N}} \frac{\Delta f_j^t}{\alpha}$ is expected to be $O(1)$, and thus computationally conducive.

We remark that in Alg. 3, other than linear update rule, non-linear rules (e.g., quadratic) can also be used to adjust the pace of updating. Also, instead of the parameter α , a specific α_i can be specified for each bulb $i \in \mathcal{M}$.

5 TESTBED IMPLEMENTATION

As a proof-of-concept of the proposed smart lighting control system, we implemented a prototype testbed using programmable smart bulbs and smartphones as light sensors, which was deployed in a real-world indoor environment. This section elaborates on the testbed setup and implementation details.

²By drawing from a uniform distribution on $[0, 1]$ and then computing the negative logarithm.

³The O^* notation hides poly-logarithmic factors.

Algorithm 3 ContinualAdjust $[(f_j^{t-1}(x^{t-1}))_{j \in \mathcal{N}}, (f_j^t(x^{t-1}))_{j \in \mathcal{N}}]$

```

1: for  $j \in \{1, \dots, n\}$  do
2:    $\Delta f_j^t \leftarrow \min \left\{ f_j^t(x^{t-1}) - f_j^{t-1}(x^{t-1}), 0 \right\}$ 
3:   if  $\Delta f_j^t + f_j^t(x^t) \geq 1$  then
4:      $\Delta f_j^t \leftarrow 0$ 
5:   end if
6: end for
7:  $x_i^t \leftarrow x_i^{t-1} \quad \forall i \in \mathcal{M}$ 
8: repeat
9:   for  $i \in \{1, \dots, m\}$  do
10:     $x_i^t \leftarrow x_i^t + \alpha \cdot \max_j |\Delta f_j^t|$ 
11:   end for
12:   if  $f_j^t(x^t) \geq 1$  then
13:      $\Delta f_j^t \leftarrow 0$ 
14:   end if
15: until  $\max_j |\Delta f_j^t| = 0$ 
16: return  $(x_i^t)_{i \in \mathcal{M}}$ 

```

5.1 Testbed

The deployed testbed system encompasses 9 LIFX LED smart light bulbs, 3 Samsung Nexus 5 smartphones and a laptop computer, as illustrated in Fig. 4. The light bulbs are actuated by a central controller, implemented on a laptop computer using the Flask library of Python programming language. The controller communicates with the light bulbs and sensors via a local wireless network. Each bulb can deliver up to 1100 lumens with power consumption around 11W. The proposed lighting control algorithm is implemented within the central controller as a Python program. A simple smartphone application is developed with Android programming language, which broadcasts the local illuminance levels recorded at sensors to the central controller, with a frequency of 1s. These readings serve as an input for the proposed control algorithm.

5.2 Deployment Environment

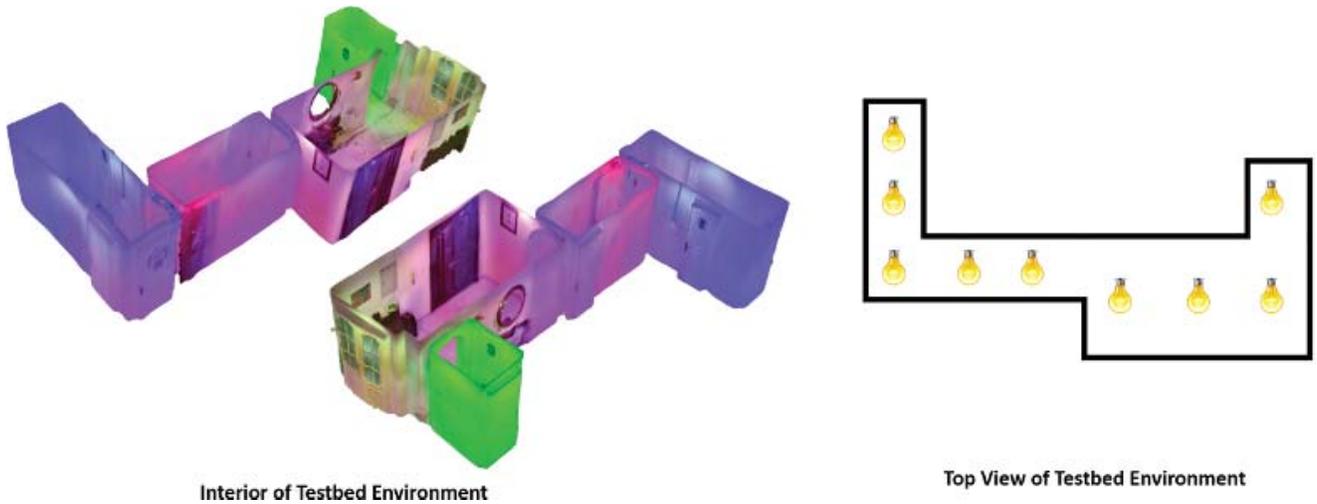
The testbed environment is illustrated in Fig. 3. The demonstration site occupies a portion of the first floor of a residential apartment, where the light bulbs were evenly installed in 3 semi-enclosed areas divided by open doors and short hallways as illustrated in the rightmost part of Fig. 3. The site area is approximately $18.6m^2$ with a ceiling height of $2.5m$. This environment presents heterogeneous lighting conditions in a sense that some light bulbs may cross-illuminate while others may not.

6 PERFORMANCE EVALUATION

This section presents the results of experiments conducted, which appear in Figs. 5 and 6. In the subsections to follow, we list the experimental scenarios considered, adopted experimental settings and explain the results.

6.1 Experimental Scenarios and Settings

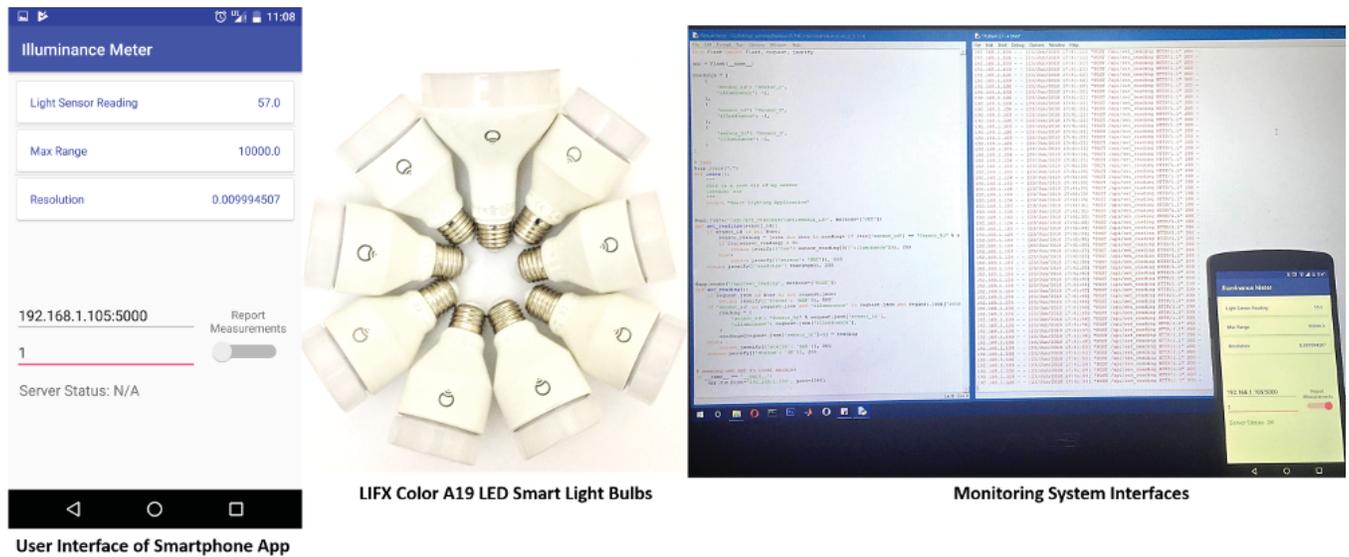
For a comprehensive evaluation of the proposed algorithm, diverse settings and scenarios were considered taking into account users'



Interior of Testbed Environment

Top View of Testbed Environment

Figure 3: 3D and 2D illustrations of the testbed environment. The locations of installed 9 LIFX smart luminaries are marked with icons in the top view plan.



LIFX Color A19 LED Smart Light Bulbs

Monitoring System Interfaces

User Interface of Smartphone App

Figure 4: Components of the deployed smart lighting testbed: Graphical user interface of the developed smartphone application, LIFX Color LED smart light bulbs and monitoring system.

mobility and lighting preferences, number of sensors as well as light bulbs. Particularly, the following are settings for the case studies performed.

- a) Experiment 1 (E1): In this scenario only 1 bulb and 1 sensor were considered. The user’s lighting preference was held constant throughout the experiment, while the location was varied dynamically.
- b) Experiment 2 (E2): Here, 1 sensor and 2 bulbs were considered. The user’s lighting preferences and location were varied over time.

- c) Experiment 3 (E3): Similar to E2, but with 2 bulbs and 2 sensors.
- d) Experiment 4 (E4): This scenario considers a larger scale instance with 9 light bulbs and 3 sensors. Users’ locations were fixed but lighting preferences were altered over time.

In all the experiments performed, the smartphones were kept on a surface facing towards the ceiling although not necessarily directly under a light bulb. We remark that using smartphones as illuminance sensors was a matter of availability rather than a constraint of this experimental setup. In fact, we argue that the setup would yield similar results, if not better, with wearable sensors

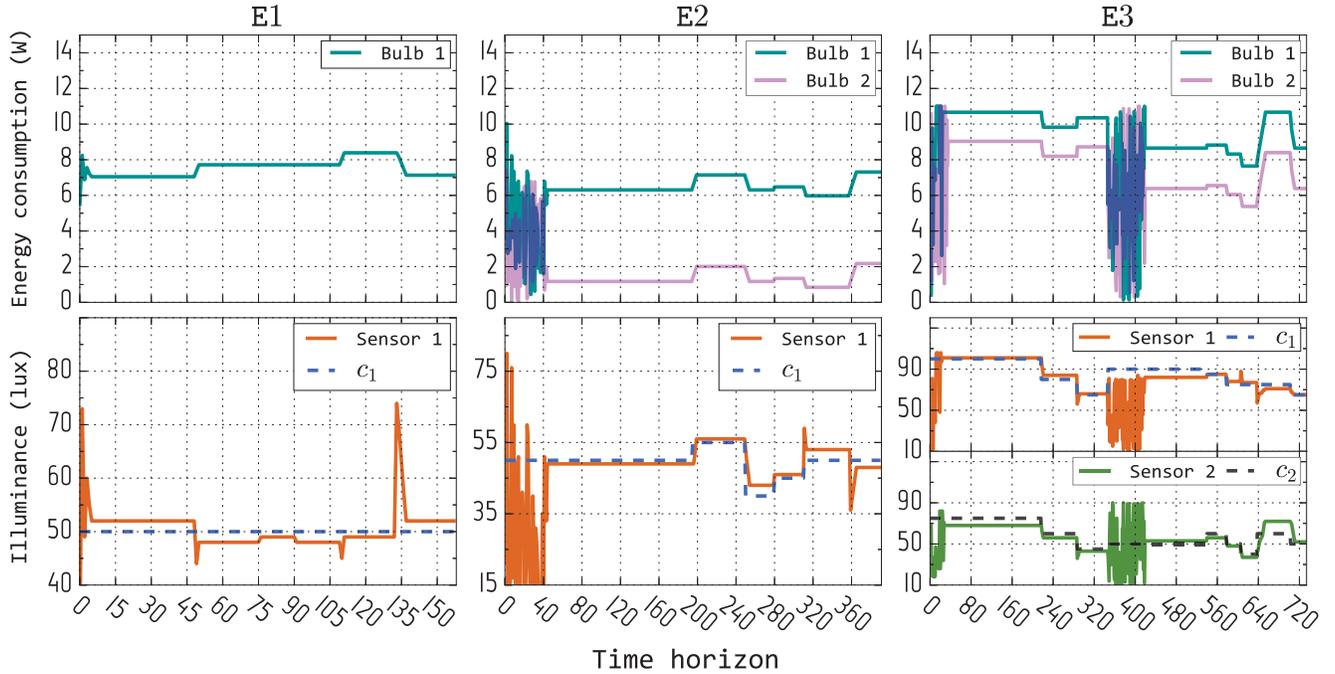


Figure 5: Energy expenditure of light bulbs, users' perceived and preferred (c) illuminance levels over time for experiments E1, E2 and E3.

(e.g. Google Glass, smartwatch, etc.) as such devices capture users' lighting preferences more accurately.

The deployed monitoring unit in the proposed setup executes either the bootstrapping control to learn the environment or the continuous adjustment algorithm to meet users' lighting requirements rapidly. A major change detected between user's reported preference and the recorded illuminance at the corresponding sensor will invoke the bootstrapping algorithm in order to relearn the environment. Conventionally, this threshold is defined to be strictly greater than 20 lux, below which the continuous adjustment algorithm should be adequate.

For practical purposes, the update step size parameter τ in Alg. 2 was set to 0.4; a decision made under extensive empirical experimentation. Also, the parameter η , which characterizes the extent of feasibility of produced solutions, was set to 0.3 (when $\eta = 0$, the solutions are exactly feasible).

6.2 Results

The proposed algorithm proved successful in meeting users' lighting preferences while minimizing the net energy expenditure of smart bulbs, as could be inferred from Figs. 5 and 6. Despite the active mobility of users in almost all the experiments, the algorithm successfully induced continuous adaptive control of light bulbs in real-time. For example, as illustrated in Fig. 5, at around time instant 45 in experiment E1 user suddenly changed the current location distancing away from the light bulb. This resulted in lower illuminance readings at the sensor, thereby triggering the controller unit to initiate Alg. 3 (continual adjustment algorithm).

Since the realized deviation in recorded illuminance was small (i.e., less than 20 lux), the latter algorithm was invoked instead of Alg. 2 (bootstrapping control).

Though Alg. 3 is efficient for small lighting perturbations, it might fail to properly minimize the energy wastage or meet users' heterogeneous preferences at the events of significant lighting variations in the environment. For example, in experiment E3 at time instant near 360 Alg. 2 is invoked since user 1's lighting preference increases from 65 to 90, as exemplified in Fig. 5. Also, the bootstrapping algorithm is particularly efficient when the experimental environment is not yet explored, as the case is in the beginning of experiments. This is because Alg. 2 requires no apriori knowledge concerning the bulbs or sensors, whereas Alg. 3 requires as an input the current energy consumption levels of bulbs. Therefore, in all the conducted experiments the bootstrapping algorithm is invoked first.

However, a major downside of the bootstrapping algorithm is attributed to the sampling subroutine invoked during the environment exploration stage. Even with smooth lighting transitions enabled, abrupt large fluctuations in the brightness levels of light bulbs may cause considerable discomfort for users. Importantly, it is the diverging magnitude of these oscillations that inflicts the discomfort, rather than the fluctuations themselves. Throughout the sampling phase, Alg. 2 generates random points from the entire simplex region, as explained in Sec. 4, and thus bulbs' brightness levels are expected to frequently oscillate between 0% and 100% instead of converging. As this impairs practicality of the developed control framework in some applications, as is the case with smart

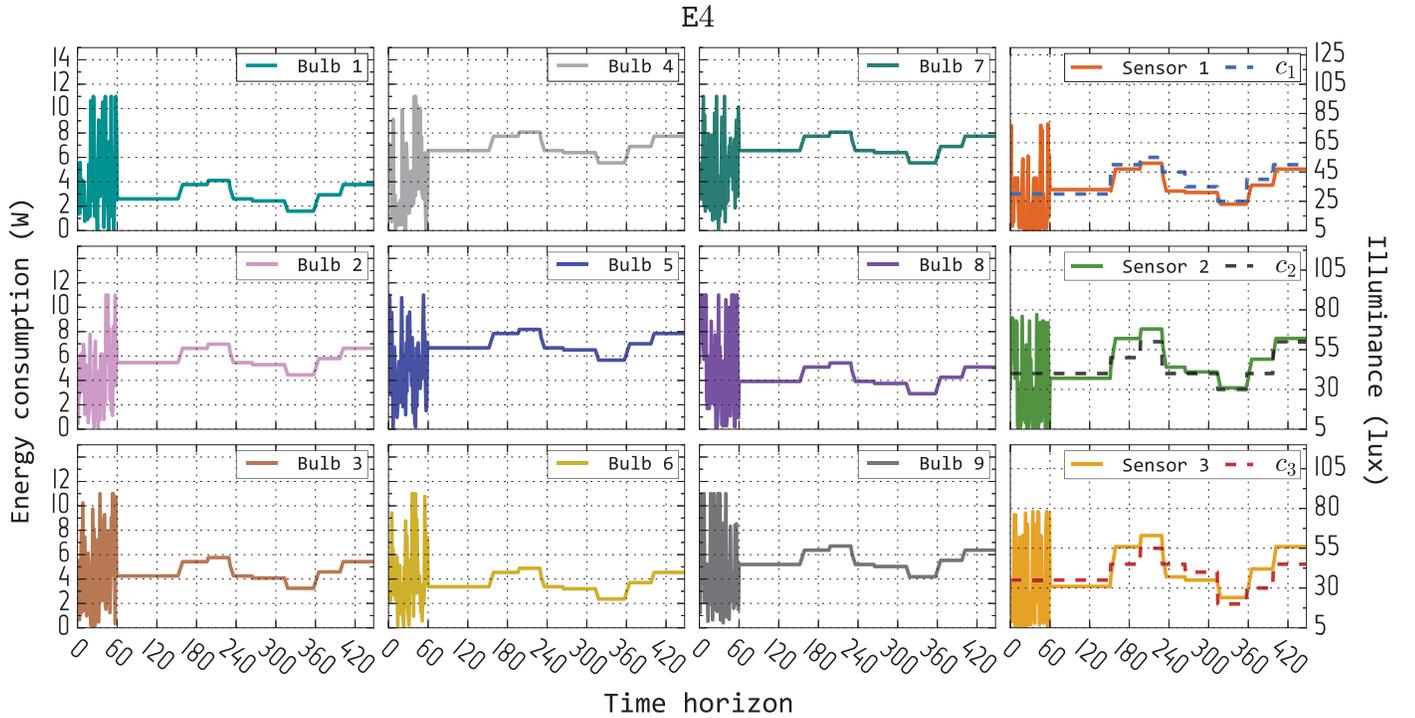


Figure 6: Energy expenditure of light bulbs, users' perceived and preferred (c) illuminance levels over time for experiment E4.

lighting systems, it is vital to minimize the intensity of fluctuations thereby assuaging users' visual discomfort. The following section circumvents this concern by providing efficient disturbance minimizing sampling strategies, which will be implemented in future work.

Lastly, experiment E4 seeks to validate the scalability of the proposed approach on a larger scale instance with 9 bulbs and 3 sensors. As could be seen from Fig. 6, the employed approach satisfied users' heterogeneous requirements consistently without noticeably over- or under-illuminating. However, the bootstrapping phase witnessed a longer period of fluctuations in E4 than in E1, E2 and E3, signifying that the algorithm explored a larger sampling space. Despite this, the featured control algorithm managed to optimize light bulbs' brightness levels within plausible time frame, thus confirming its scalability in larger problem spaces.

7 DISCUSSIONS

While the deployed testbed prototype demonstrates the empirical effectiveness of the proposed smart lighting control approach, it should be noted that a further enhancement is required to tackle the practical limitation mentioned previously. This section discusses the latter along with the future applications of the presented smart lighting control system.

7.1 Disturbance Minimization

In Section 3, the problem of oblivious multi-sensor control was reduced to an equivalent one with a known feasibility region, consequently enabling a more efficient and rapid sampling during the bootstrapping phase. Yet, this stage could be still disconcerting as, to the users, numerous large fluctuations of light bulbs translate to an unsettling period of random flickering, hence to visual discomfort and disturbance.

In response to this hindrance, several potential solutions are explored below that minimize the inflicted disturbance to users.

- (1) *Improved Sensors and Light Bulbs*: The implemented testbed prototype relies on Android smartphones and LIFX smart bulbs connected through a wireless local network. Whereas, it is possible to design specific wearable devices with improved light sensors and dedicated communication network that have faster reaction time and refined communication capabilities. This will minimize the latency in data transfer between sensors, smart bulbs and monitoring unit. Moreover, future smart bulbs are expected to be equipped with improved power electronics allowing rapid brightness adjustments, which will further reduce the control lags.
- (2) *Optimized Sequence of Random Samples*: Rather than relying on hardware improvement, we can extend the current bootstrapping control algorithm with a *disturbance minimizing sampling routine*. The underlying idea is to truncate the sampling space during each iteration of the while loop in Alg. 2. To achieve this, first, the feasibility region (i.e.,

the simplex $S(B)$ is partitioned into a preset number of roughly equal parts. Then, in each iteration one of the parts is discarded, based on a certain rule, thereby reducing the overall size of the sampling domain. Importantly, this allows to retain the efficient uniform sampling technique (with only slight modification) leveraged by Alg. 2 and at the same time diminish the magnitude of lighting fluctuations. Furthermore, instead of probing the uniformly generated brightness levels at a random order, they can be sorted to minimize the flashing effect between each consecutive pair of samples (e.g., by choosing the sample nearest to the current one).

7.2 Color Adjustments

The present study empirically demonstrates the effectiveness of intelligent lighting control system in a real-world environment. However, our control system can be also applied to a more sophisticated setting of multi-colored illuminance control [3]. For example, the users can set heterogeneous preferences on the *color temperature*. Different colors exhibit different radiation effects. Reddish colors exhibit warmer effect, while bluish colors colder effect. Color temperature is conventionally expressed in kelvin (K), a unit of measure for absolute temperature. In particular, it is well-known that warmer colors are more pleasant to humans in dim environment, whereas colder colors are better for bright environment. Also, different users may have different psychological preferences to color temperature. Hence, given heterogeneous preferences of color temperature, our goal is the coordinate the color and brightness adjustments of smart light bulbs.

8 CONCLUSION

This paper considers a practical application of IoT for smart lighting systems, whereby mobile light sensors (possibly integrated in wearable devices like smart watches and smart spectacles) are utilized to measure the local illuminance in different areas. Upon receiving the illuminance measurements, a set of smart light bulbs will be able to coordinate to automatically adjust their brightness levels to minimize the total energy consumption, while meeting the local acceptable illuminance constraints. The challenges for such a system include the uncertainty of the locations of sensors and light bulbs, as well as time-varying background light sources. We solved an *oblivious multi-sensor control problem*, whose objective is to enable effective control without complete knowledge of the dynamic uncertain environment, subject to certain quality-of-service

constraints. We provided a viable solution is based on learning and random sampling by a subset of measurements.

REFERENCES

- [1] Muhammad Aftab, Sid Chi-Kin Chau, and Peter Armstrong. 2013. Smart Air-conditioning Control by Wireless Sensors: An Online Optimization Approach. In *Proceedings of ACM International Conference on Future Energy Systems (e-Energy)*. 225–236.
- [2] Muhammad Aftab, Chien Chen, Chi-Kin Chau, and Talal Rahwan. 2017. Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system. *Energy and Buildings* 154 (2017), 141–156.
- [3] Simon HA Begemann, Ariadne D Tenner, and Gerrit J Van Den Beld. 1998. Lighting system for controlling the color temperature of artificial light under the influence of the daylight level. (Feb. 24 1998). US Patent 5,721,471.
- [4] Dimitris Bertsimas and Santosh Vempala. 2004. Solving convex programs by random walks. *Journal of ACM* 51, 4 (2004), 540–556.
- [5] Chi-Kin Chau, Majid Khonji, and Muhammad Aftab. 2016. Online Algorithms for Information Aggregation From Distributed and Correlated Sources. *IEEE/ACM Transactions on Networking* 24, 6 (Dec 2016), 3714–3725.
- [6] M. D. Grigoriadis, L. G. Khachiyan, L. Porkolab, and J. Villavicencio. 2001. Approximate Max-Min Resource Sharing for Structured Concave Optimization. *SIAM Journal on Optimization* 11, 4 (2001), 1081–1091.
- [7] Adam Kalai and Santosh Vempala. 2006. Simulated Annealing for Convex Optimization. *Mathematics of Operations Research* 31, 2 (2006), 253–266.
- [8] M. T. Koroglu and K. M. Passino. 2014. Illumination Balancing Algorithm for Smart Lights. *IEEE Transactions on Control Systems Technology* 22, 2 (March 2014), 557–567.
- [9] László Lovász and Santosh Vempala. 2004. Hit-and-run from a corner. In *ACM Symposium on Theory of Computing (STOC)*. 310–314.
- [10] László Lovász and Santosh Vempala. 2006. Fast Algorithms for Logconcave Functions: Sampling, Rounding, Integration and Optimization. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. 57–68.
- [11] Dennis E. Phillips, Rui Tan, Mohammad-Mahdi Moazzami, Guoliang Xing, Jinzhu Chen, and David K. Y. Yau. 2013. Supero: A Sensor System for Unsupervised Residential Power Usage Monitoring. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*.
- [12] F. Rubinstein, M. Siminovitch, and R. Verderber. 1993. Fifty percent energy savings with automatic lighting controls. *IEEE Transactions on Industry Applications* 29, 4 (Jul 1993), 768–773.
- [13] Vipul Singhvi, Andreas Krause, Carlos Guestrin, James H. Garrett, Jr., and H. Scott Matthews. 2005. Intelligent Light Control Using Sensor Networks. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*.
- [14] Noah A Smith and Roy W Tromble. 2004. Sampling uniformly from the unit simplex. *Johns Hopkins University, Tech. Rep* 29 (2004).
- [15] Yao-Jung Wen and A. M. Agogino. 2008. Wireless networked lighting systems for optimizing energy savings and user satisfaction. In *IEEE Wireless Hive Networks Conference*. 1–7.
- [16] Y-J. Wen and A. M. Agogino. 2011. Control of wireless-networked lighting in open-plan offices. *Lighting Research & Technology* 43, 2 (2011), 235–248.
- [17] Nan Zhao, Matthew Aldrich, Christoph F. Reinhart, and Joseph A. Paradiso. 2015. A Multidimensional Continuous Contextual Lighting Control System Using Google Glass. In *ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys)*. 235–244.