Design and Implementation of a Wireless Sensor Network for Intelligent Light Control

Heemin Park and Mani B. Srivastava
Networked and Embedded Systems Laboratory
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA 90095
Email: {hmpark, mbs}@ee.ucla.edu

Jeff Burke
Center for Research in Engineering, Media and Performance
University of California, Los Angeles
Los Angeles, CA 90095
Email: jburke@remap.ucla.edu

ABSTRACT

We present the design and implementation of the Illuminator, a preliminary sensor network-based intelligent light control system for entertainment and media production. Unlike most sensor network applications, which focus on sensing alone, a distinctive aspect of Illuminator is that it closes the loop from light sensing to lighting control. We describe the Illuminator's design requirements, system architecture, algorithms, implementation and experimental results. To satisfy the high-performance light sensing requirements of entertainment and media production applications, the system uses the Illumimote, which is a multi-modal and high fidelity light sensor module well-suited to wireless sensor networks. The Illuminator system is a toolset to characterize the illumination profile of a deployed set of fixed position lights, generate desired lighting effects for moving targets (actors, scenic elements, etc.) based on user constraints expressed in a formal language, and assist in the set up of lights to achieve the same illumination profile in multiple venues. After characterizing deployed lights, the Illuminator computes at run-time optimal light settings to achieve a user-specified actuation profile using an optimization framework based on a genetic algorithm Uniquely, it can use deployed sensors to incorporate changing ambient lighting conditions and moving targets into actuation. With experimental results, we demonstrate that the Illuminator handles various high-level user's constraints and generates optimal light actuation profile. These results suggest that our system should support entertainment and media production applications.

1. INTRODUCTION

Wireless Sensor Network (WSN) technologies have enabled many interesting applications in pervasive and ubiquitous computing [18]. UCLA's history of interdisciplinary research among engineering, media, performance and other arts provides an opportunity to explore application of WSN technologies to entertainment, multimedia and media production [3]. The Advanced Technology for Cinematography (ATC) [28] project, a collaboration between the Networked and Embedded Systems Laboratory (NESL) and the Center for Research in Engineering, Media and Performance (REMAP), is exploring wireless sensor network support of filmmaking. The ATC project seeks to enhance entertainment production and provide both increased expressive capabilities and significant cost savings by deploying sensor networks on film sets. In the ATC project, we initially focused on capturing and archiving sensory data from the set into a database in frame-rate synchronization using sensor network technologies. To do so, we developed the Augmented Recording System (ARS) [26] and demonstrated the possibilities and usefulness of the wireless sensing in filmmaking. Although the ARS itself provides many benefits, it is a one-way data collection application. On the film set or in theater, there are many types of equipment to be controlled by crews such as lights, audio and cameras, many of which use advanced digital control systems but do not incorporate sensing. We wanted to explore the possibilities of using WSN technologies to actuate and control such equipment, not just monitor it. Because lighting is vitally important in film and theater while being relatively straightforward to control, we sought to develop an intelligent light control system for entertainment and media production using wireless sensor networks.

1.1 Intelligent Light Control System for Entertainment and Media Production

Computerized control systems for lights in film and theater are available as commercial products [4, 7]; but most of these current systems only provide for actuation and do not exploit sensor data. We believe that it is important to know and use the live light information from light sensors deployed on the set. Real-time data accounts for how characteristics like light intensity and color temperature change over time and deployments due to filament aging, supply voltage variation, changes in fixture position, color filters, etc. Without real-time measurement of light, it is time-consuming to maintain desired intensities of lights for certain area across many venues and over long time periods. Light intensities and color temperature can be measured accurately by the currently available handheld manual light meters [12, 20]. However, these devices have not been incorporated in systems supporting automatic light control and must be manually moved through different points in space. Cameras can provide only reflected light intensity, so we fo-

IPSN '07 Cambridge, MA, USA

cus on incident light in order to have measurements that are independent of surfaces and materials. We have developed an intelligent light control system, Illuminator, which finds and manages the best light actuation profiles using incident light measurements by light sensors and user requirements.

1.2 Roles of the Intelligent Light Control System

The Illuminator is a toolset to help media production crews characterize, control and setup lights in performance and filmmaking using sensor network technologies. The Illuminator has three roles: *given a light setup and user constraints, 1) recommend sensor deployment, 2) characterize the lights, and 3) based on light characteristics and sensor readings, find and manage the best light actuation profiles satisfying user constraints.* These constraints represent requirements about lighting's aesthetic effect and include desired light intensities of the field and/or high level description of lighting conditions; they are discussed in detail in Section 3.1. We assume that lights have fixed position over the time of Illuminator's control. Although we could use pan-tilt mounts and automated lighting that can move the direction of lights and control color, we did not consider these features for our system. Tracking and spotlighting using pan-tilt mounts is a well-known technology and can be implemented easily [25]. We do allow mobile stage elements, equipment and actors lit by these fixed lighting instruments, using mobile tags. Tag is a single entity that can sense light intensity and know its location. Brief descriptions of the Illuminator's three roles follow.

Sensor Placement Recommendation: The Illuminator recommends the best sensor locations using a limited number of available light sensors. Uniform and regular sensor placement will work well for general cases. However, if user has specific light intensity pattern that they want to generate, the sensor placement should change according to intensity patterns. The Illuminator system parses user constraints and shows the recommend sensor locations in a GUI display on a host workstation.

Light Characterization: To generate desired light levels at specific locations, knowledge of the projection pattern of lights and brightness according to dimmer level is required. We call this information the *Light Characteristics* and the process of capturing this information is *Light Characterization.* Enabled by the additive characteristics of light intensity and for simplicity, the characterization process is done by turning on each light one by one at each dimmer level and measuring the incident light intensities using wireless light sensors.

Optimal Light Actuation Profile Generation: This is the core of the Illuminator light control system. The Illuminator system finds the best light level for each light to meet user's requirement with the measured light characteristics information. The algorithms and management of the user requirements and actuation profile are described in Section 3 and Section 4.

In addition to these roles, the Illuminator can be used for reconstructing similar lighting effects in a different physical light setup as well. This requires re- characterization of the lights. Though time-consuming to execute, this feature becomes very powerful, for example, when the same performance needs to be done in different places or at different times. If lights are re-setup or setup in different places, the setup will vary even though the crew tries to set it up in the same way. We use Illuminator to document the results of a lighting setup, not just its physical assignment.

1.3 Design Requirements

Although the design driver for the Illuminator is entertainment and media production, our system could also be used for other purposes, such as achieving a required illumination profile in outdoor public venues, high-rise office buildings, retail spaces, etc. Design requirements for applications in entertainment and media production are however the most challenging among those applications.

High Fidelity and Network-Capable Light Sensing: The first necessary technology to realize a sensor-supported light control system for entertainment and media production is high performance light sensing. In WSN, the Mica node [5] is the de facto standard platform to form wireless networks. Therefore, a high fidelity and network- capable light-sensing module for Mica platform will offer tremendous advantages. We have developed a multi-modal and high-fidelity light sensing module, the Illumimote [17]. The Illumimote is well-suited for light control applications. It provides multi-modal light sensing capabilities (e.g. incident light intensity and RGB color intensities for color temperature), wide dynamic range and fast response time. Illumimote provides the critical sensing substrate for the Illuminator system.

Understanding High-Level User Constraints: Expected users of the Illuminator light control system are the persons who are in charge of setting, controlling and managing lighting of film sets or theaters. Even for them, it is hard to quantify the requirements and constraints of what a field's desired illumination. Lighting design is typically an iterative, hands-on process. At a minimum, it's desirable that the users describe their intentions in high-level terms familiar to them; thus the Illuminator should understand and implement high-level user constraints. Examples of high-level user constraints include: "Maintain 1000 lux brightness at location (10, 20)", "Keep illuminating a moving actor with 1000 lux brightness", "Illuminate an actor with 500 lux brightness from front side and 1000 lux from left side", "The difference of illumination on location A and B should be 3000 foot-candle", "Illuminate an actress with 4:1 contrast ratio from left side to right side", and "Evenly illuminate over an area". User constraints include desired illumination and associated time and locations.

Adaptive to Environment: In many cases, ambient light from uncontrollable light sources is present in the field. The Illuminator should support closed-loop control methods that incorporate real-time light measurements from sensors.

1.4 Contributions and Paper Organization

This paper describes the design requirements and necessary technologies for light control systems based on wireless sensor network technologies, especially for entertainment, multimedia and media production. Our work provides following contributions. 1) We address unique design requirements and implementation issues for wireless sensor networks in an application that brings together sensing and control. 2) We identify the required processes and methods for intelligent light control using wireless sensor networks: light characterization, parsing of user constraints, optimal light actuation profile generation, and closed-loop control with real-time feedback that accommodates changing ambient lighting conditions and moving targets. A language for user constraints is defined and the light control problem is formulated as an optimization problem. Based on the identified processes, the language for user constraints and developed methods, we provide a framework for light control for entertainment and media production. 3) We verify the capabilities of our system with a proof-of-concept testbed. The rest of this paper is organized as follows. In Section 2, related work with light control is described. Problem formulations and approaches of the light control system are presented in Section 3. In Section 4, system architecture and implementation issues are described. In Section 5, experimental setup and results of our implemented light control system are presented. Section 6 discusses the future work and concludes.

2. RELATED WORK

Cast Software provides commercial lighting software, Wysiwyg [4] which is a 3D CAD tool for designing lighting for theater and performance. This tool supports virtual view and rendering of theatrical lighting in 3D, including both fixture deployment and light profile. Electronic Theatre Controls, Inc. provides a computerized control system, the Eos lighting console [7] which can control up to 5000 channels (devices). However, these products rely on theoretical light characteristic information for each fixture from prebuilt libraries when these can be more accurately obtained by measuring the actual field. The tool and control console cannot utilize active sensing and closed loop control. UCLA's ARS [26] provides a framework to monitor and archive sensor data from the set. The data can be used at shooting for decision support for cinematographers such as directors, script supervisors and directors of photography (DPs). Also, the data can be utilized for *continuity management* and for synthesizing computer generated image (CGI) with live actor in post- production. Similarly, REMAP has applied wireless sensing technologies to theater and media production. Burke designed an interactive system for a production of *Macbett* [2], in which a custom software uses wireless sensing devices to dynamically control lighting and sound effects based on performers' position and movement. The sensor technology used in [2] was the MLD tracking system [16] which uses four beacon speakers and ultrasound for distance measurement. Elsewhere, for multimedia, the SEVA system [14] used WSN to determine if objects are in the view field of camera. Video footage is annotated with the object IDs and later the IDs are used to search video stream which contains that objects. SEVA uses the locations of objects obtained using MIT's Cricket ultrasonic localization nodes. The ACCESS project [21] spotlights an audience using a pan-tilt mount, which is also a capability of the MLD and several other commercial systems. The location is computed using a video camera and image processing. An example of using WSN for coordinated light control for building illumination was presented by Singhvi et al. [23], who formulated the light control problem as a trade-off between building occupant's utility function and energy consumption. This utility function indicates the level of satisfaction for each building occupant. However, it does not support representation of illumination quality for cases involving multiple sites, which happens frequently in illumination for entertainment and media production purposes. Also, Singhvi et al. assume that the light level at each point of interest from each light source is known; but they did not show how this information is obtained and validated. Illuminator provides a systematic process to obtain this information via a light characterization phase.

3. PROBLEM FORMULATIONS AND APPROACHES

Based on the roles of a closed-loop light control system described in Section 1.2, the following subsections describe the problem formulations and approaches of the major tasks.

3.1 Representation of User Constraints

In theatrical and film lighting design, expression of user's intention (constraints) may be very complex and hard to describe in terms of absolute brightness. This is because the intention of lighting tends to involve multiple points and conditions. For example, in film, a director of photography (DP) may want equal illumination on two points, different illuminations on two points with 200 lux brightness difference, or illumination on two points with contrast ratio of 3:1. We use these examples as an initial set of high-level constraints from which to develop the Illuminator System. To provide users the capability of expressing such high-level constraints about lighting and at the same time, to provide the light control system a way to understand the constraints, we defined a simple but formal language for representation of user constraints. We categorized user's intention about the lighting into six types and the language was designed to provide those six lighting criteria.

1) ABSOLUTE Type: This is to specify brightness of certain area or tags by absolute value in lux or foot-candle.

2) GREATER/LESS Type: This type is used to specify minimum/maximum brightness of area or tags.

3) DIFF Type: This type is to specify relative brightness between two points For example, this is used when illumination on some points need to be brighter/darker by absolute amount of brightness (e.g. 500 lux) than other points. In conjunction with `GREATER/LESS` type, the minimum or maximum of the difference can be specified.

4) RATIO Type: This is used when user wants to maintain a consistent contrast ratio on certain points (e.g. actor's face) for lights from two different directions or different lights, e.g., 3:1 ratio from the left side to right side of the actor. This type of constrain can also be used to specify relative brightness in percentage. For example, if one location needs to be illuminated 50 % brighter than other point, it can be represented by the contrast ratio 1.5:1.

5) EQUAL Type: User may want even illumination over an area or a set of points. Because the `EQUAL` type constraints are supposed to find an illumination for evenness, not finding illumination of particular brightness, this is different from the case of specifying all area to be illuminated by same brightness with `ABSOLUTE` type constraints.

Regardless of the constraint types, the Illuminator system should provide consistent illumination with or without external uncontrollable lights (e.g. sunlight). Additionally, the tags on actors or props may be moving and the Illuminator system should change light actuation profile to adapt to environment as the tags move.

We defined a language that can express the preceding six types of user's interests. Fig. 1 shows the grammar of the language. The **lux** and **fc** (foot-candle) are units of light

Constraint = <ID>**:** **illuminate** [Brightness **at**] ([**mobile**] Tag | Area) **from** (**all** | Lights) [Condition]+ ;
| <ID>**:** **illuminate** **map=**<filename> **from** (**all** | Lights) [Condition]+ ;
| <ID>**:** **set** Relation [Condition]+ ;
| <ID>**:** **place** Area ;
| <ID>**:** **monitor** Area ;
Brightness = Number Unit
Number = <Integer> | <Real>
Unit = **lux** | **fc**
Tag = **tag** <Integer>
Area = **all** | Point | Polygon
Point = "(" Number **,** Number")"
Polygon = "(" Point **,** Point (**,** Point)+ ")"
Lights = "(" Light (**,** Light)+ ")"
Light = **light** <Integer>
Relation = Constraint **−** Constraint **=** Brightness
| Constraint **/** Constraint **=** Number
| Constraint **>** Brightness
| Constraint **<** Brightness
| Constraint (**=** Constraint)+
Condition = **during** Interval | **when** Tag **in** Area
Interval = "[" Time **,** Time "]"
Time = Integer [**ms** | **sec**] | **end**

Figure 1: Grammar for the High-Level Constraints

intensity and the ratio of foot-candle to lux is 10.76 [29]. The language also has keywords **place** and **monitor** to indicate preferred sensor locations, and area of interest for monitoring, respectively. With the language, the area can be specified by points or polygon.

Each constraint is activated by two conditions: time period and location. Except for the `EQUAL` type, each constraint has an associated objective value *gamma* which is the value to be achieved when finding optimal light actuation profile for the constraint. Here are some examples of user constraints, their types and *gamma* values.

`C1:illuminate map=leftspot.bmp from all;`
(Type: `ABSOLUTE`, γ values are obtained from the map file.) This constraint represents that user wants to have an illumination as similar as the map file which is named `leftspot.bmp` using all lights all the time.

`C2:illuminate 1000 lux at tag 0 from all`
`during [500ms, 2500ms];`
(Type: `ABSOLUTE`, $\gamma = 1000$) This constraint represents an illumination of 1000 lux brightness on tag 0 from all lights during 500 millisecond through 2500 millisecond.

`C3:illuminate 50 fc at mobile tag 1 from`
`(light0, light1) when tag 1 in (20, 30);`
(Type: `ABSOLUTE`, $\gamma = 538$) This represents an illumination of 50 foot-candle (= 538 lux) on tag 1 only from light 0 and light 1 when the tag 1 is at location (20, 30). Light id is determined by the dimmer channel connected to the light. The foot-candle unit is converted into lux.

`C4:illuminate tag 2 from (light0, light1);`
`C5:illuminate tag 2 from (light2, light3);`
`C6:set C4 - C5 = 100 lux;`
(Type: `DIFF`, $\gamma = 100$) This constraint represents an illumination with relative brightness. The illumination on tag 2 from light 0 and 1 needs to be brighter than that from light 2 and 3 by 100 lux.

`C7:set C4 / C5 = 3;`
(Type: `RATIO`, $\gamma = 3$) This means the ratio of illumination on tag 2 from light 0 and 1 to that from light 2 and 3 is 1:3.

`C8:set C4 > 300 lux;`
(Type: `GREATER`, $\gamma = 300$)
This constraint limits the minimum brightness to 300 lux.

`C10:illuminate tag 3 from all;`
`C11:illuminate tag 4 from all;`
`C12:illuminate tag 5 from all;`
`C13:illuminate tag 6 from all;`
`C15:set C10 = C11 = C12 = C13;` (Type: `EQUAL`)
These constraints are for even illumination. Constraint $C15$ means the illuminations on tag 3, 4, 5, and 6 need to be as equal as possible. The `EQUAL` type constraint does not need γ value.

After parsing the constraints, type, objective value γ, associated locations or tags, and associated lights of each constraint are stored in a text file in the database directory and then the light actuation profile generation task uses this file.

3.2 Sensor Placement Recommendation

Most previous work on sensor placement deals with coverage issues by regular placement and deployment for better estimation of unknown fields [6,8,19,30]. Variance-based approaches are used for better estimation of the field by adopting adaptive and incremental scheme to find next placement locations of high variance (or entropy). In this case, the variance is computed based on the measurements by the currently placed sensors. Unlike previous research on sensor placement, in our application the user knows what the resultant light field should be like. Our system suggests sensor placement for verification of whether the intended light field is properly created.In our system, we combine the two typical approaches to these problems: the regularity-based technique and variance-based technique. Sensor placement is calculated in an incremental fashion. For sensor placement locations, we want place the next sensor at a location that is the farthermost point from the current placement in terms of both of distance for regularity and variance. In our case, the variance can be computed between the desired intensities. Let Y be the possible placement locations and let $P = \{p_1, p_2, p_3, ..., p_n\}$ be the current placement. $D(P, y)$ is minimum combined distance of Euclidean distance and distance in their desired value (light intensities) from new location y to the current placement P. Then, $D(P, y)$ is defined in equation 1. The combined distance $D(P, y)$ represents how farther the sensor at location y is from the current placement P.

$$D(P, y) = \min_{\forall p_i \in P} \{d(y, p_i) + K \cdot |I(y) - I(p_i)|\} \qquad (1)$$

where $d(y, p_i)$ is the Euclidean distance between y and p_i, K is the weight for linear combination of two terms (we set $K = 0.03$), and $I(y)$ and $I(p_i)$ are the desired light intensities at location y, p, respectively. The two terms represent regularity and variance by the distance and the difference of the desired intensities between y and the closest location in P, respectively. The next best sensor location would be y that maximizes $D(P, y)$. The incremental placement selection algorithm is shown in Fig. 2. The sensor placement recommendation only considers `ABSOLUTE` type of constraints. If there are multiple constraints, the maximum variance across the constraints is used. Sensor deployment

```
$P = \emptyset$
$Y =$ all possible locations
for number of available sensors begin
    $\hat{y} = \operatorname{argmax}_{y \in Y} D(P, y)$
    $P = P \cup \hat{y}$
    $Y = Y - \hat{y}$
end
```

Figure 2: Distance-Variance Based Placement Algorithm

is recommended by displaying the optimal location of the sensors and the current deployment on a GUI display, providing real-time feedback of the current deployment.

3.3 Light Characterization

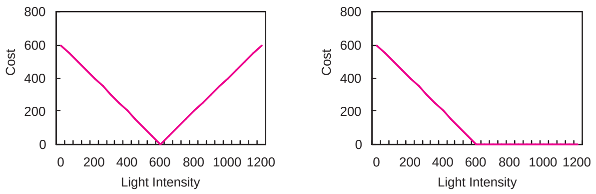
To compute the intensity of illumination at a given location, the transfer function from all possible combinations of dimmer settings and locations to light intensities must be known. This knowledge is equivalent to the *Light Characteristics* for a given lighting deployment and is a function of dimmer setting and target location. Let $\lambda : X \times Y \rightarrow I$ be the light characteristics, where X is the set of possible light actuation profiles (combinations of settings of dimmers), Y is the set of locations of interest, and I is the generated light intensity. Then, the straightforward light characterization process involves $\lceil \frac{n(Y)}{s} \rceil \cdot d^l$ steps, where l, d and s are the number of lights, dimmer levels and available sensors, respectively. This is of exponential complexity and thus practically infeasible.

Fortunately, it is already known that light intensity is additive [23] and we exploit this property for the light characterization process. Because the total incident light intensity at a location is summation of light intensities at that point from all sources illuminating the location, we can characterize lights one by one. We sweep through all intensities for a given light and measure light intensities. Then, we can redefined the light characteristics function as $\lambda : L \times D \times Y \rightarrow I$, where L is the set of lights, and D is the set of possible different levels of a dimmer. With this characterization process, the complexity becomes $\lceil \frac{n(Y)}{s} \rceil \cdot ld$ and this is practically feasible. We assume that the area is divided into fixed-size grids. For other locations not characterized by the sensors, we estimated intensities by applying Natural Neighbor Interpolation scheme [22] an interpolation technique based on Voronoi diagrams.

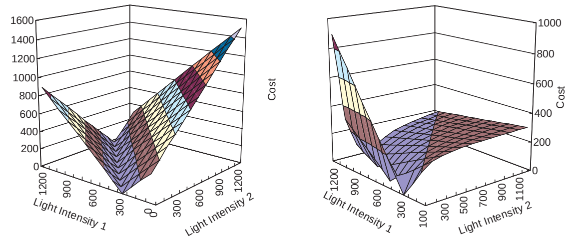
3.4 Light Actuation Profile Generation

Generating optimal light actuation profiles (value for each dimmer over time) for user constraints is the main role of the Illuminator system. There are typically many lights creating the field. Therefore,light intensity at a location is affected by multiple sources and the lighting should be done in coordinated fashion.

The inputs for light actuation profile generation are the light characteristics λ (Section 3.3) and user constraints (Section 3.1), and output is the optimal light actuation profile (dimmer intensity values) that satisfies user constraints. Let $\mathbf{x} = \{x_1, x_2, x_3, ..., x_n\}$ be a light actuation profile, which is a set of dimmer intensity values, where n is the number of lights. Then, the incident light intensity $I_y^{\mathbf{x},L}$ at location y from lights L with light actuation profile $\mathbf{x}$ can be computed using equation 2.



(a) $\phi_{\texttt{A}}$: `ABSOLUTE` Type (b) $\phi_{\texttt{G}}$: `GREATER` Type



(c) $\phi_{\texttt{D}}$: `DIFF` Type (d) $\phi_{\texttt{R}}$: `RATIO` Type

Figure 3: Cost Functions of Each Constraint Type

$$I_y^{\mathbf{x},L} = \sum_{\forall i \in L} \lambda(i, x_i, y) \qquad (2)$$

We defined a set of cost functions $\Phi = \{\phi_{\texttt{A}}, \phi_{\texttt{G}}, \phi_{\texttt{L}}, \phi_{\texttt{D}}, \phi_{\texttt{R}}, \phi_{\texttt{E}}\}$ to indicate how well the lighting matches with the user constraints. Examples of the cost function $\phi \in \Phi$ of constraint type `ABSOLUTE`, `GREATER`, `DIFF` and `RATIO` are shown in Fig. 3. Letter `A`, `G`, `L`, `R`, `D`, and `E` represents `ABSOLUTE`, `GREATER`, `LESS`, `RATIO`, `DIFF` and `EQUAL` type, respectively, Fig. 3(a) and 3(b) show the cost functions when the required brightness γ is 600 lux, Fig. 3(c) shows the cost when required brightness difference of two illuminations is 300 lux and Fig. 3(d) shows that of desired contrast ratio being 3:1. The cost functions for `ABSOLUTE`, `GREATER` and `LESS` type have two arguments because they compare generated light intensities and desired intensities. The `DIFF` and `RATIO` type have three arguments as these functions calculate the difference and ratio of the first two input arguments, then compare the result with the third value. The `EQUAL` type may have more than two arguments, as this type assesses evenness of the light intensities across many points. The evenness is determined by computing the standard deviation of the light intensities at the locations of interest.

Let $c \in C$ be a user constraint and $\{y_c^1, y_c^2, ..., y_c^m\}$, γ_c and L_c be the associated sensor locations, objective value and associated lights of the constraint c, where m is the number of locations associated with constraint c. Constraint type `DIFF` or `RATIO` has two associated locations, y_c^1 and y_c^2. Constraint type `EQUAL` may have more than two associated locations and it does not need γ input. Then, the cost for a constraint c with light actuation profile $\mathbf{x}$, $\phi(c, \mathbf{x})$ can be computed as in equation 3, where *type*(c) represents the type of the constraint c.

$$\phi(c, \mathbf{x}) = \phi_{type(c)}(\{I_{y_c^1}^{\mathbf{x},L_c}, I_{y_c^2}^{\mathbf{x},L_c}, ..., I_{y_c^m}^{\mathbf{x},L_c}\}, \gamma_c) \qquad (3)$$

For total cost, we use the root-mean-square to minimize the costs of constraints altogether. Then, finding the best light

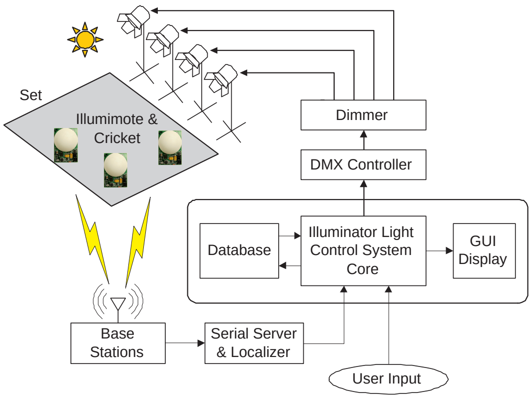


Figure 4: Illuminator Light Control System

actuation profile X for given constraints C can be formulated as an optimization problem.

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \sqrt{\sum_{\forall c \in C} \phi^2(c, \mathbf{x})} \tag{4}$$

$\hat{\mathbf{x}}$ would be the optimal dimmer values that minimizes the total cost.

We can extend this approach to allow the Illuminator to control lights under the condition of existence of uncontrollable external lights such as sunlight. The external lights are regarded as ambient light and assumed to change slowly. If the ambient light at location y at time t is $\rho_y(t)$, measured incident light intensity is $I^*_y(t)$, and actuation profile is $\mathbf{x}$, following equation holds because light intensity is additive.

$$I^*_y(t) = I^{\mathbf{x},L}_y + \rho_y(t) \tag{5}$$

Using the sensors' light measurement and estimated light intensity from the light characteristics, we can estimate the ambient light intensity $\rho_y(t)$. The $\rho_y(t)$ can be obtained by subtracting generated light intensity $I^{\mathbf{x},L}_y$ from measured light intensity $I^*_y(t)$. Then, the incident light intensity at location y with actuation profile $\mathbf{x}$ can be redefined as following equation.

$$I^{\mathbf{x},L}_y = \sum_{\forall i \in L} \lambda(i, x_i, y) + \rho_y(t) \tag{6}$$

The ambient light intensity may change over time, so whenever light measurement is updated, the $\rho(t)$ needs to be updated too. Although the ambient light changes over time and locations, it does not require re-characterization of lights as long as the light setup remains same.

4. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The entire Illuminator system can be divided into three subsystems: Sensor network, Illuminator core, and DMX controller and dimmer. Fig. 4 shows the overall system connection of the Illuminator light control system.

4.1 Sensor Network

The sensor network measures the light intensities and sensor locations. and consists of two sub-networks: one is the Cricket localization system and the other is single-hop MicaZ [5] network with the Illumimote light sensing board [17].

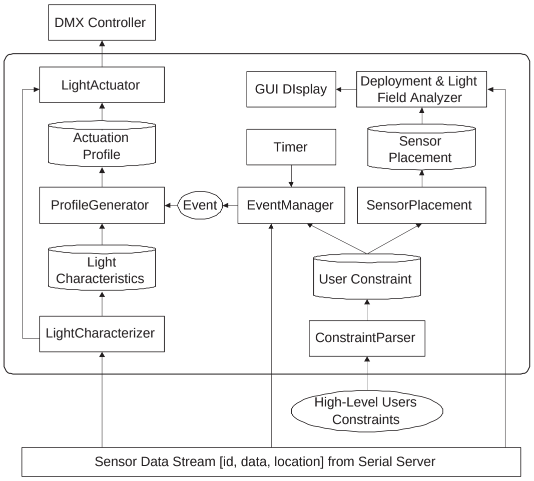


Figure 5: System Architecture of the Illuminator Light Control System Core

We used three Cricket [24] nodes for beacon nodes that are pre-calibrated with their locations, and we coupled a Cricket with each Illumimote for localizing the light sensor module. The Cricket nodes run on TinyOS [27] and measure distances between them using ultrasound. The Illumimote node runs the SOS [9] environment on a MicaZ node in order to achieve higher bandwidth. The Illumimote provides rapid and accurate measurements of incident light intensity and RGB color intensities (for color temperature). In order to manage two sensor network platforms, two Java modules run concurrently: SerialServer for interfacing between the Illuminator core and sensor networks, and Localizer for computing positions of the Cricket nodes based on the ultrasound range measurements.

4.2 DMX Controller and Dimmer

For development, we used a Leprecon 6-channel dimmer [13] and ENTTEC DMXEtherGate MK2 [1] as the DMX-controlled dimming system. (DMX refers to the entertainment industry standard control signal for lights, 8-bit dimmer levels multiplexed on an RS485 serial link.) The Illuminator core sends 521-byte control packets to the DMX controller via the Ethernet. Then, the DMX controller sets the dimmer according to the control packets. The dimmer has six 100 W channels and can control each light at 8-bit resolution.

4.3 System Architecture & Components of the Illuminator Core

The system architecture of the Illuminator core is shown in Fig. 5, with subtasks as rectangles. Modules are implemented as separate Java threads and thus run concurrently. The Illuminator keeps the intermediate data in text files in the database directory and it is shown in cylindrical shape. The Illuminator can be operated in different modes according to its role: Sensor Placement Recommendation, Deployment Assistance, Light Characterization, Off-line Profile Generation, On-line Profile Generation. Because each module is an independent thread, each operational mode only activates the necessary modules. For example, Light Characterization mode only activates the LightCharacterizer, LightActuator and SerialServer modules. The On-line Profile Generation mode activates the ProfileGenerator, LightActuator, EventManager, Timer, ConstraintParser and SerialServer modules.

4.3.1 ConstraintParser

We implemented the ConstraintParser with the Java Compiler Compiler (JavaCC), a parser generator for Java [11]. Using JavaCC syntax, we implemented the grammar in Fig. 1. The parser module reads the user constraints in the form of language described in Fig. 1 and constructs the UserConstraints database in tuples of [type, activation conditions, associated tags and locations, associated lights, γ value].

4.3.2 LightCharacterizer

This module controls all lights in turn for available dimmer settings. Each sensor reads the incident light intensities at its location and the sensor readings are stored in LightCharacteristics database in tuples of [light id, dimmer level, location, incident light intensity]. To consider ambient lights, each intensity reading will be subtracted by the intensity reading at the dimmer value 0 (no intensity). After measuring all locations, Natural Neighbor Interpolation scheme [22] is applied to estimate light characteristics for unsensed locations.

4.3.3 EventManager

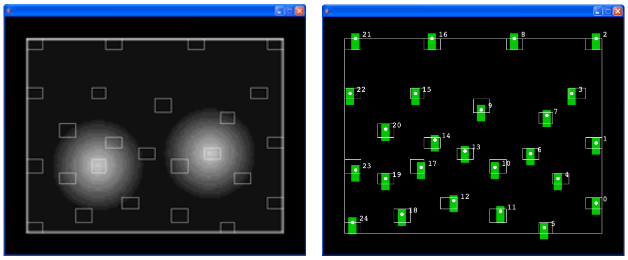
Execution of actuation profile generation is triggered by events. Events are constructed based on sensor data (e.g. location of the tag), user constraints and current time. The Timer module maintains the current time in milliseconds. The EventManger module periodically looks into the activation conditions of the UserConstraints database and checks if the light actuation profile needs to be updated by the active constraints based on current time, sensor data and locations. Then, the EventManager signals the ProfileGenerator module with the event. The event is implemented as an object which contains user constraints to be satisfied.

4.3.4 ProfileGenerator

As explained in Section 3.4, the light actuation profile is found by a combinatorial optimization framework. The ProfileGenerator module finds the best combination of dimmer values for lights to meet user's desired intensity level at the sensor locations and other requirements. We implemented a genetic algorithm [10] for the optimization method and described its implementation only briefly here. First, 100 random solutions are generated to create the initial population. In the case of on-line profile generation, the current actuation profile from the ActuationProfile database is added to the initial population to avoid search from beginning. The population's cost is computed and solutions are sorted by cost. Then, the worst 20 % solutions are removed. With the best 20 % solutions, mutation and crossover are performed and the new solutions added to the population. In crossover, dimmer values of lights are exchanged with each other. Mutation perturbs the dimmer values. Then, the cost is recomputed and above procedure repeated until satisfied solution is found or maximum number of search is reached.

4.3.5 LightActuator

Actuation Profile database is a set of tuples consisting of [start time, dimmer level for light 1, light 2, ..., light n]. LightActuator actuates lights according to this profile. LightActuator module reads the current actuation profile corresponding to the current time and sends control packets to the DMX controller. For light characterization, the LightActuator module is directly controlled by the LightCharacterizer module.



(a) Placement for Left and Right Spotlight (b) Sensor Deployment based on Recommendations

Figure 6: Intensity Map for User Constraints and Their Sensor Placement Recommendations

5. EXPERIMENTAL RESULTS

5.1 Experimental Setup

To evaluate the capabilities of the Illuminator light control system, we set up a proof-of-concept experimental setup in a small lab.The experimental setup has the same structure as shown in Fig. 4. We set up a small stage on the floor whose size is 102 cm by 82 cm, and four halogen lights were used for light sources. Four halogen lamps on the wall were connected to the dimmer: the leftmost lamp is light 1 and the rightmost lamp is light 4. (All of the following figures of the stage are drawn upside down so that lights are placed at the bottom.) Although four beacon nodes are required for unique localization in 3-D, we used only three Cricket nodes because we assumed that sensors are on the floor and thus the Z-coordinate is always zero. Because of the limited availability of Illumimote prototypes, we deployed only one Illumimote node and one Cricket node. (We slightly modified the Illuminator core and we measured multiple points to emulate multiple sensors.) We set the maximum brightness that could be specified by the bitmap image file as 2000 lux. Then, maximum variance is 2000 and maximum distance between sensors is about 130 cm. Based on this, we set K = 0.03 for the equation 1 to give regularity double weight than variance.

5.2 Sensor Placement Recommendation

We first tested the capability of guided sensor placement. We assumed that the user wants to illuminate according to two light intensity maps as shown in Fig. 10 (a) and (b): left spotlight and right spotlight. With the intensity maps, recommended sensor placement for 25 sensors was obtained. The placement recommendation results are shown in Fig. 6(a) and 6(a). To provide flexibility in physical deployment, recommended locations are shown in rectangle areas instead of points. For intensity maps of left spotlight and right spotlight, denser sensor placement is recommended for areas of light intensity changes as shown in Fig. 6(a).

5.3 Sensor Deployment and Light Characterization

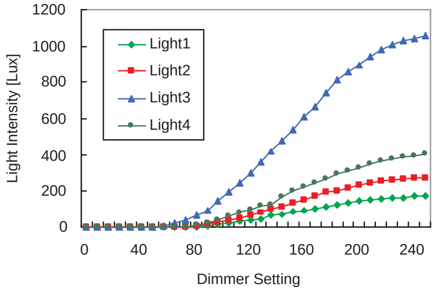
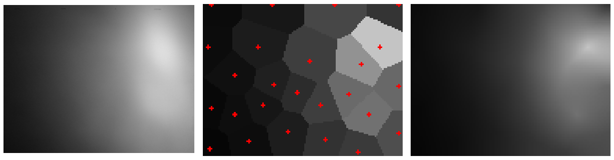


Figure 7: Light Characteristics at Sensor 14



(a) Photograph (b) Voronoi Tessellation (c) Interpolated Characteristics

Figure 8: Characteristics of Light 1 at Maximum Dimmer Level

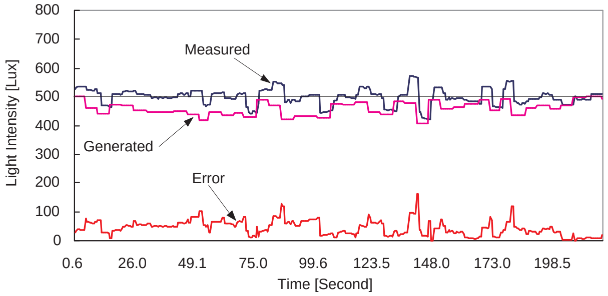
We characterized lights for 32 dimmer levels at the 25 points as recommended in Fig. 6(a). Fig. 6(b) shows the figure of the actual deployment that we characterized lights according to the recommendations. The characteristics are different for each light and for each sensor. Fig. 7 shows an example of light characteristics of sensor 14. Because the sensor 14 is deployed on the left side, we can guess it is affected from light 3 and 4 more than light 1 and 2.

We conducted this characterization process for 4 lights at each of 25 points. In order to demonstrate the quality of the light characterization results after the interpolation, we took photographs of the stage with illuminating at maximum dimmer level. For camera, a Canon 10D was used with settings of lens aperture of F/4, focal length of 17mm, and shutter speed of 1/125 second at ISO 400. Fig. 8 (a) shows the photograph of the stage with illuminating by light 1. Fig. 8 (b) shows the Voronoi tessellation based on light measurements from 25 sensors. Based on this Voronoi tessellation, Natural Neighbor Interpolation [22] was applied and Fig. 8 (c) shows the interpolated light characteristics including unsensed area.

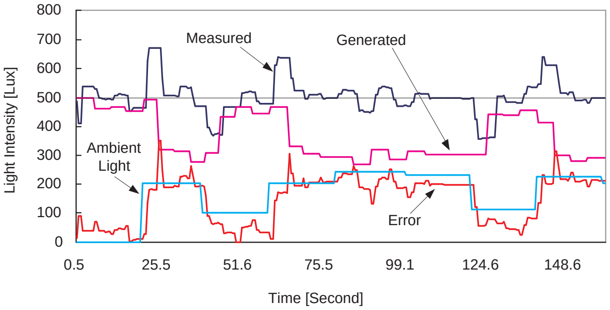
The photographs are a little bit brighter because the stage is white and thus highly reflective. The interpolated characterization results only show the incident light intensities, so they look a little darker.

5.4 Light Actuation Profile Generation

Once the light characteristics are obtained, a light actuation profile for any constraints can be generated. We tested the Illuminator system in four categories: Absolute Brightness, Even Illumination, Relative Brightness, and Contrast Ratio. For absolute brightness, five intensity patterns described by bitmap images and consistent illumination with and without random ambient light are tested. To test the capabilities of understanding various types of user constraints, even illumination, relative brightness and contrast ratio are tested as well. Because of space limitation, we cannot include the user constraints statements we used and we only show the results of Absolute Brightness, Relative Brightness, Contrast Ratio in this paper.



(a) Without Ambient Light



(b) With Ambient Light

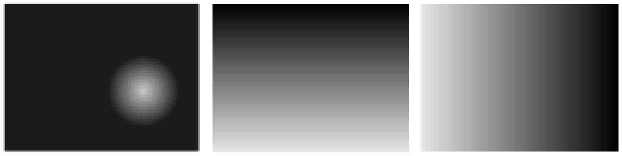
Figure 9: Experimental Results for Consistent Illumination

5.4.1 *Absolute Brightness*

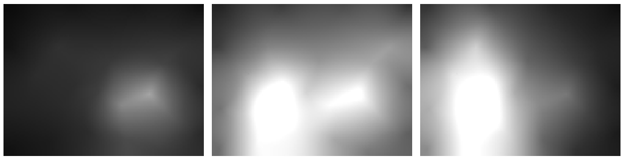
In this category, we tested to see our system can generate and maintain absolute brightness.

Consistent Illumination: We tested our system with and without random ambient light. A user constraint that required absolute and consistent illumination of 500 lux at tag 0 was used. In order to see the behavior of our system, we continued generating light actuation profile even after the brightness requirement was met. Fig. 9 shows the experimental results of the consistent illumination. The Generated Intensity curve shows the estimated brightness by the light characteristics, $I_y^{X,L}$, as described in Section 3. Measured curve shows the brightness as measured by the Illumimote light sensor. There is a difference between generated brightness and measurement, which is the error. There are several sources of this error: sensing error, interpolation error in light characteristics, localization error and ambient light intensity. As one can see in Fig. 9(a), although there are some fluctuations, our system maintains consistent brightness.

We also tested the Illuminator with such random ambient lights. To generate random ambient lighting, one additional halogen lamp is connected to fifth channel of the dimmer and set to random intensities by the Illuminator core. Fig. 9(b) shows the experimental results with the existence of such random ambient light. Please note that the unit of the ambient light in Fig. 9(b) is not lux, but dimmer value; so, it is not same as brightness in lux. Whenever there are changes



(a) User Constraints – Intensity Profiles



(b) Estimated Intensities based on Light Characteristics

(c) Photographs of the Real Illuminations

Figure 10: Experimental Results of Illumination for the Fixed Intensity Profiles

in ambient light, there are also changes in the measurement. The measured brightness converges to 500 lux. The `Error` curve appropriately tracks the ambient light changes; this means that the Illuminator adapts to environment correctly.

In the consistent illumination experiments, the optimization framework implemented using a genetic algorithm searched from 100 to 800 solutions for real-time generation of one light actuation profile. It took at most 5 millisecond on a Pentium 1.5GHz laptop computer and this is enough for the frame rate of film or video (33 millisecond).

Fixed Illumination: To evaluate the capabilities of illuminating according to intensity map images, we tested our Illuminator with five intensity maps of which size is 128 × 97 pixels and each pixel represents intensity from 0 to 2000 lux. The Illuminator can read bitmap image files that express desired light intensities on the field. Five bitmap images of Fig. 10 (a) represents user's desired light intensities. Fig. 10 shows experimental results of fixed illumination for three bitmap images: right spotlight, front light, and left light, respectively from the left columns. The Illuminator generated the light actuation profiles that match with the given bitmap images the most. With the light characteristics, generated light intensities by the Illuminator are estimated as shown in Fig. 10 (b). For comparison, we took photographs of the real illuminations by the generated actuation profile as shown in Fig. 10 (c). Like in Fig. 8, the real illumination and estimated illumination are very similar.

5.4.2 *Relative Brightness*

The capability of relative illumination was tested in this experiment. We wrote user constraints that describe illumination difference on two locations (tag 13 and tag 14) from -500 lux through 500 lux. In addition, we added constraints that illuminations should be greater than 100 lux to avoid the case of illumination on either tag being too dark. Fig. 11 shows the illumination results by our system. The plots for `C0` and `C1` represent light intensities on tag 13 and tag 14, respectively. Two light intensities `C0` and `C1` change accordingly to satisfy the illumination difference requirements.

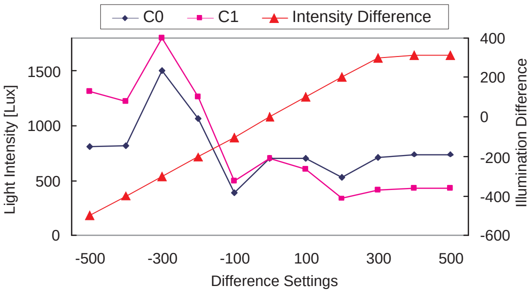


Figure 11: Experimental Results for Relative Illumination – Difference

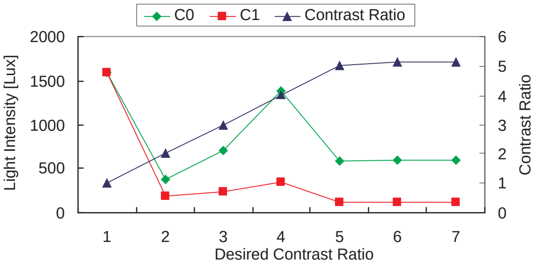


Figure 12: Experimental Results for Contrast Ratio

5.4.3 *Illumination with Contrast Ratio*

Finally, we evaluated the capability of illuminating with a varying contrast ratio. We prepared user constraints that require contrast ratio from 1:1 through 7:1 between illuminations on tag 10 and tag 14. Like experiments on the relative illumination, we added constraints that illuminations on tag 10 and tag 14 should be greater than 100 lux to avoid too dark illuminations. Fig. 12 shows the plots for `C0` and `C1` which represent light intensities on tag 10 and tag 14, respectively. To satisfy the requirements, light intensities of `C0` and `C1` change as the contrast ratio changes. The plot of the contrast ratio shows that the Illuminator generated proper light actuation profiles to meet the desired contrast ratio in most cases. For the last two cases (contrast ratio 6:1 and 7:1), the contrast ratio requirements were not satisfied. In this case, because of the proximity of the two tags, it was to hard to generate significant difference between illuminations on two points (e.g. 7 times brighter than other).

In addition to the experiments we have described so far, we performed experiments on even illumination and contrast ratio for mobile tags as well. The Illuminator found correct light actuation profiles for those cases.

6. CONCLUSIONS

We have presented system architecture, requirements and implementation of an intelligent light control system for entertainment and media production using WSN to close the loop. To satisfy the high-performance light sensing requirements, we used the Illumimote, a multi-modal and high fidelity light sensor module designed for the Mica family of wireless sensor nodes. Then, we created the Illuminator, a preliminary toolset to characterize lights, generate desired lighting effects for user constraints and help deployment lights in entertainment illumination applications. We developed a simple language to formally describe users' high-level constraints. An optimization framework was applied to find optimal light settings with given light setups. In experimental results, the Illuminator was shown to appropriately handle various high-level user constraints, optimal light actuation profile generation in on-line and off-line fashion. More information including source code and images of the experimental results are posted in the project website, http://nesl.ee.ucla.edu/research/Illuminator.

Immediate future work includes the deployment of the Illuminator in a sound stage or theater. Additionally, although the Illuminator can adopt any localization system, Cricket nodes were used because of their simplicity and availability. For future work, Vanderbilt's Radio Interferametric Positioning Systems [15] or commercial localization systems may be used to extend the scale of this system and accuracy of the sensor locations. Applying more sophisticated multi-input multi-output control algorithms would be another area of future work. Currently, the Illuminator uses optimization techniques to find the best light setting and relies on the light characteristics. This is appropriate when the number of lights and number of dimmer levels are small. However, when the number of lights increases to hundreds or more, the optimization framework will not work well for on-line and real-time light control. It would be desirable to approach the light control problem from a control theoretic perspective where a priori knowing light characteristics information is not needed. This would be useful when light characteristics are not accurate (e.g. for interpolated area) or not available.

Finally, the Illuminator's initial development focused on controlling only light intensity. Modern lighting control systems, especially in theatrical and high-end architectural applications, are capable of changing color, beam spread and other characteristics on-the-fly. Future versions of the Illuminator may explore how to handle color and color temperature. The Illumimote light sensing module we have developed already has the capability of sensing RGB color separately and color temperature sensing. Therefore, the Illuminator can extend the characterization process to RGB color space and the control algorithm can be extended to three color channels.

7. REFERENCES

[1] DMX512 Ethernet Gateway MK2 User Manual. http://www.enttec.com/docs/ethergate_manual.pdf.

[2] J. Burke. Dynamic performance spaces for theater production. *Theatre Design & Technology (U.S. Institute of Theatre Technology)*, 38(1), 2002.

[3] J. Burke *et al.* Embedding expression: Pervasive computing architecture for art and entertainment. *Journal of Pervasive and Mobile Computing*, pages 1–36, February 2006.

[4] Cast Software. Wysiwyg. http://www.castlighting.com.

[5] CrossBow. Mica data sheet. http://www.xbow.com.

[6] S. Dhillon and K. Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *WCNC 2003*, 2003.

[7] ETC. Lighting control products. Website, http://www.etcconnect.com/products.controls.asp.

[8] C. Guestrin *et al.* Near-optimal sensor placements in gaussian processes. In *ICML*, pages 265–272, 2005.

[9] C.-C. Han *et al.* A dynamic operating system for sensor nodes. In *MobiSys*, 2005.

[10] J. H. Holland. Genetic algorithms. *Scientific American*, 267:44–50, 1992.

[11] Java Compiler Compiler. http://javacc.dev.java.net.

[12] Konica Minolta. Minolta Color Meter IIIf. http://konicaminolta.com.

[13] Leprecon. Leprecon ld-360 users manual. Website, http://www.leprecon.com/ProductSupport/LD-360_Dimmers/21-1044c_360MPX_manual.pdf.

[14] X. Liu *et al.* Seva: Sensor enhanced video annotation. In *Proceedings of ACM Multimedia*, November 2005.

[15] M. Maróti *et al.* Radio interferometric geolocation. In *SenSys*, 2005.

[16] Martin Lighting Director. http://www.martin.com.

[17] H. Park *et al.* Illumimote: Multi-modal and high fidelity light sensor module for wireless sensor networks. *IEEE Sensors Journal*, 2006. in press.

[18] G. J. Pottie *et al.* Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[19] M. Rahimi *et al.* Adaptive sampling for environmental field estimation using robotic sensors. In *IROS*, 2005.

[20] Sekonic. Sekonic L-558Cine DualMaster. http://www.sekonic.com/Products/L-558Cine.html.

[21] M. Sester. ACCESS Project. http://www.accessproject.net.

[22] R. Sibson. *A brief description of the natural neighbor interpolant, In: D.V. Barnett, editor, Interpreting Multiariate Data*. John Wiley & Sons, 1981.

[23] V. Singhvi *et al.* Intelligent light control using sensor networks. In *SenSys*, 2005.

[24] A. Smith *et al.* Tracking Moving Devices with the Cricket Location System. In *Mobisys*, 2004.

[25] Spotlight. Website, http://www.spotlight.it.

[26] N. M. Su *et al.* Augmenting film and video footage with sensor data. In *PerCom*, 2004.

[27] TinyOS. Website, http://www.tinyos.net.

[28] F. Wagmister *et al.* Advanced technology for cinematography, 2002. Website, http://hypermedia.ucla.edu/projects/atc.php.

[29] Wikipedia. Lux. http://en.wikipedia.org/wiki/Lux.

[30] R. Willett *et al.* Backcasting: adaptive sampling for sensor networks. In *IPSN*. ACM Press, 2004.