Vision-based User-centric Light Control for Smart Environments

Huang Lee, Chen Wu, and Hamid Aghajan

Wireless Sensor Networks Lab
Department of Electrical Engineering
Stanford University, Stanford, CA 94305
{huanglee,chenwu,aghajan}@stanford.edu

Abstract

Smart homes are conceptualized as environments responsive to user’s presence and actions, and adaptive to user preferences and context. Visual information plays an enabling role in smart home applications such as interfaces and gesture control. This paper reports on the use of cameras and a distributed processing method for automated control of lights in a smart home. The proposed optimization formulations maintain the user’s comfort while reducing the energy cost of lights. Information from camera sensors provides occupancy reasoning and human activity analysis. By detecting the user’s position and activity, and employing light utility functions as constraints, the system optimizes the light setting for the user’s satisfaction in the occupied area. Additionally, to reduce the transmission energy and bandwidth, we also propose an ellipsoid approximation (EA) method for occupancy reasoning by a camera network. By applying the EA method, the network nodes can aggregate the observed data progressively and maintain a fixed transmitted data length. Simulations on the occupancy results from experiments are provided to verify the proposed algorithms.

Key words: Smart Environments, Energy-Efficient Light Control, Camera Sensor Networks

1. Introduction

In smart environments, the current generation of smart light control systems aims to provide the user with a comfortable environment while minimizing energy consumption. A user-centric design notion is essential to offer a better user experience. This work provides a methodology and an embodiment for light control based on user context by considering the following objectives. First, the system needs to determine the user’s location and adjust the light setting accordingly for the area occupied by the user while turning off unnecessary lights in the unoccupied areas. Second, the system needs to understand the user’s
activities in the environment in order to provide suitable light settings for different scenarios. Furthermore, the user’s lighting preferences can be specified in the system. For example, various intensity and color combinations of lighting can be used to create different ambiences depending on the time of the day or during special activities such as watching movies, dining, studying, etc.

To achieve a finer granularity of the user’s location and activity, we consider a light control system based on the observations made by a camera network. Such a system consists of camera sensors for data collection, and a central unit or a base station in charge of data fusion and light level optimization. In a wireless camera network, distributed image processing is essential to avoid a large communication bandwidth and an extensive processing load on the base station, both results of transmitting raw video from the cameras. Therefore, useful features from image data are extracted locally in the camera nodes. Such features from each camera are then collected and fused by the base station. The information gathering process is over wireless channels usually via multi-hop communications. Hence, each camera node should be able to aggregate and combine its own data with those from other nodes to save transmission energy and bandwidth. The base station is able to determine the user’s location and activity based on the features from all the cameras. It then optimizes the light setting and sends corresponding control signals to the lights.

In this paper, we first propose a data aggregation method for occupancy sensing to reduce the transmission energy and bandwidth for the nodes. The base station then estimates user’s location and activity in the room. Based on the occupancy and activity information, we propose two light control algorithms that optimize user comfort and energy cost of the lights.

The remainder of the paper is organized as follows: Section 2 reviews the related work. In Section 3, we describe the system setup and assumptions. Section 4 and 5 introduce the occupancy reasoning and activity analysis methods, respectively. We present the light control algorithms in Section 6. Section 7 contains the performance evaluation and finally Section 8 offers concluding remarks.

2. Related Work

Light control using sensor networks has been reported in the literature. For example, [1] proposed multi-agent systems for commercial light control. In [2], light control algorithms were introduced to balance user comfort and operation costs. In [3], the authors described a light controller for theater art based on indoor positioning using a wireless sensor network. The system architecture, requirements, and implementation of a light control system for entertainment and media production were discussed in [4]. In [5], the authors considered light control for individual activities assuming each user carries a wireless illumination sensor. Most of the above works use illumination sensors to provide feedback for light control. Our work focuses more on the user. Therefore, instead of using illumination sensors, we use camera sensors to obtain the user’s position and activity as contextual clues for light utilization.
Our approach is composed of a vision processing module and an optimization module. In the vision processing module, we adopt a silhouette-based 3D shape reconstruction scheme for occupancy reasoning. Silhouette-based 3D shape reconstruction has revealed great potentials in a number of problems [6][7]. For example, the paper [8] reconstructed 3D visual hulls of a user and then fitted a model to obtain the user’s posture. In [9], the authors introduced real-time counting and people tracking by camera networks. In [10], a camera selection method was proposed based on the size of the reconstructed shape. In addition, it has been shown that voxel construction of human motions can be implemented for real-time operation [11].

Human activity analysis has been extensively studied, yet it remains a challenging task due to factors such as complexities in dealing with a high-dimensional and articulated human body model, varying imaging conditions, and the intrinsic complex nature of activities. Numerous approaches have been investigated to serve different applications. In [12], Moeslund et al. provided a good review for various levels of activity recognition and the corresponding methods. Multi-view camera setups have been studied to reconstruct the person’s 3D pose [8][13]. However, many approaches face the same problems of ambiguous image features (unless very restricted experiment environment is set) and estimation in a high-dimensional space. Some researchers have used conditional random fields (CRF) for activity recognition. In [14], Sminchisescu et al. used conditional models to classify activities such as walking, jumping, running, picking, or dancing and demonstrated that the contextual information helps to resolve ambiguities of similar gestures in the activity sequences. They experimented with both 2D image features as well as 3D body joint motion data. In this paper, since transmitting raw images is impractical in wireless networks, we develop a CRF-based activity analysis method that uses only the extracted features from each image frame.

Instead of camera sensors, other types of sensors can be used for human activity analysis. In [15], the authors studied home activity recognition with various sensors including built-in wired sensors, motion-detection sensors and RFID tags. They used decision trees as classifier and compared performance with different sensor modalities. Some researchers also proposed using wearable sensors for activity analysis. In [16], the authors used wearable vision and audio sensors to recognize the places a user visits. Instead of vision and audio sensors, the authors in [17] used tri-axial accelerometers to build a real-time system for recognizing body activities such as walking, sitting, and lying. In their system, the AdaBoost algorithm was used to achieve high classification accuracy. Furthermore, the feasibility of long-term daily activity recognition using wearable sensors was proved [18] where the authors recorded 10 hours data containing three activities performed by one user with three sensors separately attached to the body.

Compared to alternative types of sensors, we believe that through offering services such as the one presented in this work, the value of having sensors with rich information such as cameras will be seen by the users. The benefit of cameras is that they can serve several applications while other simple sensors
cannot provide information rich enough to serve multiple applications. In a practical system, several applications such as security when the user is not home, and light control when the user is at home can run, further motivating the use of cameras. In this work, each camera sensor processes images locally, therefore the images are killed at the source and privacy is maintained.

3. System Model and Assumptions

3.1. System Setup

We assume that calibrated camera sensors and a few base stations are deployed in a building as shown in Fig. 1. The camera sensors periodically make observations and send processed information to one of the base stations. The base station then uses the information to make occupancy reasoning and activity analysis for users in each room. Compared to the camera nodes, base stations are usually more computationally powerful and more expensive. Therefore, a low cost system only has a few base stations installed, and each base station is responsible for adjusting the lights for several rooms. Once the base station learns a user’s position and activity in a room, it can adjust the light setting of that room. For example, turning on more lights in the area around the user or turning off lights in other areas. Or, dimming the lights for users who are resting and brightening the lights for users who are reading. We consider two types of lights. For the first type, we assume that the light intensity can be adjusted continuously. For the second type, we assume that the lights can only be turned on or off. Two different light control algorithms will be introduced for these setups, respectively.

The vision processing algorithm flowchart is shown in Fig. 2. This flowchart describes the relationship between different functions including image process-
ing, voxel construction (occupancy sensing), activity analysis, and light control in our system. First, each sensor node conducts image processing to obtain the bounding boxes of the moving foreground. The voxel construction algorithm uses the bounding boxes from all camera sensors to estimate user height and positions. The height and bounding boxes are then used in the activity analysis module. Finally, the light control algorithm adjusts light setting based on the user position and activity. Details of the image processing module will be explained in the following sub-section. The other functional blocks will be presented in the subsequent sections.

3.2. Image Processing

Each camera runs background subtraction to obtain the moving foreground. The bounding box of each foreground segment is then extracted. Since the light setting is changing over time, an illumination-invariant background subtraction algorithm is required. Such algorithms have been proposed by several researchers. In [19], a real-time foreground segmentation algorithm with robustness to illumination changes was proposed. A homomorphic filtering based change detection algorithm was proposed in [20] to detect moving objects from light-varying monocular image sequences. In [21] multiple views were used for fast illumination-invariant background modeling. A background subtraction algorithm with adaptive density estimation was proposed in [22], which functions well with gradual and sudden illumination changes.

In this paper, we used the algorithm introduced in [22] for background subtraction. Each camera implements only background subtraction with adaptive density estimation to retrieve the bounding boxes of the users. Adaptation rate is specified through the magnitude of motion in the current frame, such that when the user moves, the background updates fast to absorb changes in the background, while when the user stays still the background updates slowly, in order to keep the static user in foreground. In practice, the adaptive-rate background subtraction works quite robustly.
To reduce false detection, we further used two Support Vector Machine (SVM) classifiers to remove bounding boxes that are too large or too small. The parameters of the SVM classifiers were trained as follows. Take one SVM for example. First, we collected around 2000 image frames from our videos that contain different activities. We used the background subtraction algorithm to create bounding boxes on each frame and then manually tagged each bounding box as valid or invalid (e.g., too small). Each bounding box presented a training data set sample \((w_i, h_i, c_i)\) where \(w_i\) and \(h_i\) are the width and height of the bounding box and \(c_i = 1\) or \(-1\) indicates whether the bounding box is valid or not. Given a data set of size \(P\), we can find the SVM parameters \(a_1\), \(a_2\), and \(b\) by solving

\[
\begin{align*}
\text{minimize} & \quad a_1^2 + a_2^2 \\
\text{subject to} & \quad c_i(a_1 \cdot w_i + a_2 \cdot h_i - b) \geq 1, \quad i = 1, \ldots, P.
\end{align*}
\]

After obtaining the parameters, we can test new data \((w_i, h_i)\) by calculating \(a_1 \cdot w_i + a_2 \cdot h_i - b\). If the result is larger than zero then it is taken as a correct detection. Otherwise it is regarded as a false detection and discarded.

4. Occupancy Sensing

The base station performs the occupancy reasoning operation based on the bounding box information contributed by the cameras. If we back project the bounding boxes from the camera’s image plane onto the space, each bounding box then forms a pyramid in space. Each intersection of these pyramids is a polyhedron, and such a polyhedron in the 3D space indicates the space being potentially occupied.

To find the exact occupied space requires high computation power and is time consuming. Therefore, we sample the space and only the sampled points are checked for occupancy. During the initialization step, the space is discretized into points. We then check if a point is in the polyhedron, in which case, the point is labeled as occupied.

4.1. Ellipsoid Approximation Method

The base station performs occupancy reasoning for a room based on the bounding boxes received from camera observations in that room. The bounding boxes require only a small amount of data to be represented. However, one drawback of this type of data is that it cannot be progressively aggregated among the sensor nodes as they take turns to broadcast their data, hence resulting in energy and bandwidth bottleneck nodes in large wireless networks. In order to reduce the amount of transmission data, we can combine the bounding box data from cameras in the same room. The camera nodes in a room can first exchange observations locally and then send the aggregated data to the base station via multi-hop communication as the scenario shown in Fig. 1.

We propose an ellipsoid approximation (EA) method to realize data aggregation for occupancy sensing. In the EA method, a polyhedron in the occupancy
space is approximated by an ellipsoid, and each node sends ellipsoids instead of bounding boxes to the base station. We use a 2D example to illustrate this idea. Consider a three-camera network as shown in Fig. 3 where the yellow ellipsoid (the ellipsoid in Fig. 3 (a)) represents an object. Assume that each camera sensor knows its coordinates and orientation. The first sensor computes a pyramid based on the observation and sends the pyramid information to sensor 2, which in turn combines the two pyramids and forms a polyhedron as shown in Fig. 3 (a). Given the polyhedron, an ellipsoid is computed to approximate the polyhedron and is sent to sensor 3 as shown in Fig. 3 (b).

After sensor 3 receives the green ellipsoid, it generates a new ellipsoid (the red middle ellipsoid in Fig. 3 (c)) that approximates the intersection of the received green ellipsoid and the pyramid from its own observation. Sensor 3 then sends the red ellipsoid to the next sensor until the information reaches the base station. Since the observations can be aggregated, each node sends equal amount of data and therefore no node dominates the energy consumption and bandwidth. Fig. 3 (d) shows the (red) ellipsoid approximation of the blue polyhedron that depicts the intersection of the three pyramids from each sensor.

Detailed steps of generating an ellipsoid are explained as follows. First, we introduce the mathematical definition of a polyhedron

\[ P = \{ x | a_j^T x \leq b_j, j = 1, \ldots, m \}, \]

where \( a_j \) and \( b_j \) are \( 3 \times 1 \) vectors that describe the boundary of the polyhedron. The definition of an ellipsoid is given by

\[ E = \{ x | (x - c)^T E^{-2} (x - c) \leq 1 \}, \]

where \( E \) is a \( 3 \times 3 \) symmetric and positive definite matrix and \( c \) is the center of the ellipsoid.
The main steps in the EA method are approximating a polyhedron (or a pyramid) by an ellipsoid and approximating the intersection of an ellipsoid and a pyramid by a new ellipsoid. Without loss of generality, we introduce the algorithm that finds an ellipsoid \( \hat{E} \) to approximate the intersection of a polyhedron \( P \) and an ellipsoid \( E \). The first step is to discretize the observed space into grid points. Each grid point is described by its coordinates represented by a 3 \( \times \) 1 vector \( g_i = [x_i, y_i, z_i]^T \). We then find a set of grid points that are in the set \( P \) and \( E \)

\[
G = \{ g_i | (g_i - c)E^{-2} (g_i - c) \leq 1, a_j^T g_i \leq b_j, j = 1, \ldots, m \}.
\]  

(4)

Given the grid points \( g_i \in G \), a node can generate a new ellipsoid by the following steps. First, we find the center of these points

\[
\hat{x} = \frac{1}{|G|} \sum_{i} x_i, \quad \hat{y} = \frac{1}{|G|} \sum_{i} y_i, \quad \hat{z} = \frac{1}{|G|} \sum_{i} z_i,
\]

(5)

and define \( \hat{c} = [\hat{x}, \hat{y}, \hat{z}]^T \). We then form a matrix

\[
A = \frac{1}{|G|} \begin{bmatrix}
\sum_{i} (x_i - \hat{x})^2 & \sum_{i} (x_i - \hat{x})(y_i - \hat{y}) & \sum_{i} (x_i - \hat{x})(z_i - \hat{z}) \\
\sum_{i} (x_i - \hat{x})(y_i - \hat{y}) & \sum_{i} (y_i - \hat{y})^2 & \sum_{i} (y_i - \hat{y})(z_i - \hat{z}) \\
\sum_{i} (x_i - \hat{x})(z_i - \hat{z}) & \sum_{i} (y_i - \hat{y})(z_i - \hat{z}) & \sum_{i} (z_i - \hat{z})^2
\end{bmatrix},
\]

(6)

and apply eigenvalue decomposition on the matrix \( A \)

\[
A = VDV^T,
\]

(7)

where the diagonal terms of \( D \) are the eigenvalues and each column of \( V \) is the eigenvector of \( A \).

Given the matrices \( D \) and \( V \), we can define a symmetric and positive definite matrix by

\[
\hat{E} = (4VDV^T)^{1/2},
\]

(8)

and the new ellipsoid is given by

\[
\hat{\xi} = \{ \hat{E}u + \hat{c} \| u \|_2 \leq 1 \}.
\]

(9)

The new ellipsoid is sent to the next sensor node for data aggregation. To describe an ellipsoid, we need 9 numbers for the symmetric matrix \( E \) and the vector \( c \). Assume that there are \( K_e \) occupants in a room, and each number is described by \( B_e \) bytes, the total amount of transmitted data is \( 9K_eB_e \) bytes. To compare with the case of sending bounding box data between the cameras, each bounding box can be described by 4 numbers. We assume that there are \( K_r \) bounding boxes from each camera and each number can be described by \( B_r \) bytes. Therefore, the amount of data from each camera is \( 4K_rB_r \), and for \( N \) cameras in a room the amount of data is \( 4K_rB_rN \) since the bounding box data cannot be aggregated. Therefore, when \( 9K_eB_e \) is smaller than \( 4K_rB_rN \), each sensor node can start aggregating data by the proposed EA method.
Figure 4: Comparison of HMM and Temporal CRF. The states are in white and observations are in gray.

5. Human Activity Analysis

Different human activities may have disparate temporal structures. They present various periodic properties in visual-based observations. For instance, activities such as walking and lying have image features more or less the same in each frame. Activities such as jumping or vacuum cleaning may exhibit several postures cycling in short periods. While those like pulling/dragging or picking up things from the floor are non-repetitive activities with a much shorter time span. Because of the different temporal structures, each frame has its own context when inferring the activity. For example, if we only look at an instance when the user is standing, it may be part of a walking sequence, or it can be a snapshot when he starts to pull open the fridge.

The longer-term context cannot be incorporated into a Hidden Markov Model (HMM) that has been widely applied in activity recognition, because of the assumption that given the hidden states, the observations are independent. Otherwise, the process may not be computationally tractable. A discriminative model such as conditional random field (CRF) is different in the following ways. First, HMM is directional, and the observations are independent given the states (Fig. 4), while the temporal CRF (chain CRF) is not directional, and the current state is determined by the previous state (for a bigram) and contextual observations. Second, the temporal CRF does not explicitly specify the state model. Instead, the state is highly influenced by the contextual observations because the model is discriminatively trained. Third, the variable length of context can be defined in learning the structure of the activities. Parameter estimation of the CRF model is a convex problem, for which global optimum is guaranteed. Inference can be performed with dynamic programming [23].

In this paper, we selected a small number of activities to demonstrate the concepts presented. The activities we selected represent a set of simple but frequent actions observed from typical users at home: walking, sitting (studying, watching TV), lying (on the sofa). In these settings it is likely that different light levels in each case can provide a more comfortable setting. In the CRF model, we use the bounding box and the height information obtained from occupancy sensing to classify the user’s activity into walking, sitting, lying, and unknown (denoted as generic). In Fig. 4, the activity state variable $a_t$...
can be one of \{a_{\text{walking}}, a_{\text{sitting}}, a_{\text{lying}}, a_{\text{generic}}\}, and observation \(y_t\) has two elements, the aspect ratio of the bounding box and the height of the user. The temporal CRF has no edge between \(a_t\) and neighboring observations, i.e., the context window has size 1 and the current state is independent of others given the current observation and the two neighboring states. The model parameters (conditional probability distributions) are learned from training data. During inference, with observations from occupancy sensing, activity state of a user is estimated to maximize the overall likelihood of the temporal CRF for a sequence.

6. Light Control Algorithms

We propose two control algorithms. One is intensity control and the other is switch control. For the intensity control, we assume that we can change the light intensity continuously within a certain range, while for the switch control, we assume that the lights can only be switched on or off.

6.1. Intensity Control Algorithm

Let \(\{c_l| l = 1, \ldots, L\}\) denote the variables for light intensity of \(L\) lights. Without loss of generality, we let the range of their values between zero and one, and a value equaling to one means that the light is at its full intensity while zero indicates the light is off. We can calculate the light intensity \(I(x, y, z)\) at the coordinates \((x, y, z)\) by

\[
I(x, y, z) = \sum_{l=1}^{L} c_l I_l(x, y, z) + \tilde{I}(x, y, z, t),
\]

where \(I_l(x, y, z)\) is the light intensity contributed by the light \(l\) at full intensity at the coordinates \((x, y, z)\), and \(\tilde{I}(x, y, z, t)\) is the light intensity contributed by the uncontrolled light sources. We assume that the uncontrolled light intensity is known for all time \(t\) and coordinates \((x, y, z)\), and those data are measured beforehand.

Our goal is to minimize the energy consumption of the lights while maintaining the user’s satisfaction of the light condition. In order to quantify a user’s satisfaction of the light condition, we adopt the concept of utility functions introduced in [2]. A utility function maps the light intensity to a user’s satisfaction represented by a value in a certain range. Different utility functions can be defined for different positions and activities. For example, the utility function around the desk area can be different from the utility function around the door area. Or, we can have different utility functions for sleeping and reading.

We define utility functions \(U(I(x, y, z), a(x, y, z))\) where \(I(x, y, z)\) is the light intensity from (10) and \(a(x, y, z) = \{a_1, a_2, \ldots, a_\alpha\}\) indicates the types of utility functions at coordinates \((x, y, z)\) where \(\alpha\) is the number of the types of utility functions. A higher utility function value means higher user satisfaction. We select concave forms for the utility function, and for all possible values of \(I(x, y, z)\) and \(a(x, y, z)\), the utility value is between zero and one.
From the occupancy reasoning module, we can obtain the occupied points. We then estimate the utility function values for those points in the space.

\[ U(I(x, y, z), a(x, y, z)), \forall (x, y, z) \in A, \]

where \( A \) is the set of all occupied points. An example of the utility functions for different activities considered in our experiments is given in Fig. 5. When the user is walking, we have a narrower utility function so the light strength will be closer to the middle point. When the user is sitting, we want to have a brighter environment, so the highest point of the utility function is close to the right. When the user is lying, we prefer a dim environment, so the utility function has its highest point close to the left. Finally, when the activity is other than the above three activities, we use a generic utility function that is fairly flat. We will see the light setting changing when the user has different activities in the simulations.

The above utility functions are solely used to demonstrate our algorithms in the simulations. In general, utility elicitation for practical situations is considered a difficult problem. Some discussions and promising results regarding the utility functions can be found in [2]. In our formulation we only require the utility functions to be concave. Therefore, in a real system, a user can have his own utility functions defined based on his preferences as long as the form is preserved. Automatic adaptation of utility functions to the user’s preferences based on a form of feedback is the subject of future work.

We also define the energy functions \( \{E_l(c_l) | l = 1, \ldots, L \} \) that are non-decreasing convex functions with the light intensity. In our formulation, we not only want to minimize the energy consumption of the lights but also to maintain the user’s satisfaction of the light condition. Therefore, the objective
function can be formulated by

\[
\sum_{l=1}^{L} E_l(c_l) - \delta_1 \left( \min_{(x,y,z) \in A} U(I(x,y,z), a(x,y,z)) \right),
\]

where the first term is the average energy consumption, the second term is the minimum utility value among all the coordinates, and \( \delta_1 \) is the trade-off parameter regulating the energy consumption and the minimum utility value. Furthermore, in order to make the light control results smooth, we can add one more term to the objective function

\[
12
\]

where \( \delta_2 \) is a constant with a small value and \( c'_l \) indicates the old light setting. The newly added term ensures that the difference between the new and old light intensity is small.

Given the objective function and the constraints, the light control problem can be formulated as a convex optimization problem

\[
\text{minimize} \quad \sum_{l=1}^{L} E_l(c_l) - \delta_1 u + \delta_2 \sum_{l=1}^{L} \| c'_l - c_l \|_2^2,
\]

where \( u \) is a variable which gives the minimum utility value. We also give constraints on \( I(x,y,z) \) such that the light intensity should be smaller and larger than certain values for all the positions such that \( I(x,y,z) \in [I_{\min}, I_{\max}] \).

In general, the convex optimization can be solved in polynomial time.

6.2. Switch Control Algorithm

In this method, we consider a light system in which the lights can only be switched on or off. In this case, if the number of lights is small, e.g., \( L \leq 20 \), it may be possible to search over the entire space of the \( 2^L \) possible solutions to find a solution. However, since the number of possible solutions grows exponentially with \( L \), for large systems this will be very time consuming. Therefore, we propose a heuristic algorithm for the switch type light control.

The switch control algorithm is based on the optimization formulation (14). Let \( \tilde{c}_l \) denote the solution of the problem (14). We want to find the solution \{\( d_l | l = 1, \ldots, L \)\} such that if \( d_l \) is one, the light \( l \) should be turned on, and if \( d_l \) is zero, the light \( l \) should be turned off. The proposed heuristic algorithm is shown in Algorithm 1. At the beginning, the algorithm sets variable \( b_l \) to 1 if \( \tilde{c}_l > 0.5 \) and sets \( b_l \) to 0 otherwise for all \( l \). If \( \tilde{c}_l \) is closer to 0.5, it means that we have lower confidence on the quantized value \( b_l \). Therefore, we first find the
values $|\hat{c}_l - 0.5|$ for all $l$. We then swap each bit of the solution $b_l$ (1 to 0 and vice versa) from the smallest value of $|\hat{c}_l - 0.5|$ to the largest value and check if it gives a lower cost. If the solution with the swapped value gives a lower cost and the minimum utility value is satisfied, we keep the new solution and check the next element. Otherwise, we keep the original solution. Note that the number of swaps is linear with respect to the number of lights ($L$) and not exponential in it.

Algorithm 1 Switch Light Control

**Input:** $\hat{c}_l$, $l = 1, \ldots, L$

**Output:** $d_l$, $l = 1, \ldots, L$

1: for $l = 1, \ldots, L$ do
2: if $\hat{c}_l > 0.5$ then
3: $b_l \leftarrow 1$
4: else
5: $b_l \leftarrow 0$
6: end if
7: end for
8: $\text{cost}_{\min} \leftarrow \text{inf}$
9: Sort lights by $|\hat{c}_l - 0.5|$ in ascending order with label $d(k)$
10: for light $d(k)$, $k = 1, \ldots, L$ do
11: $b_{d(k)} \leftarrow 1 - b_{d(k)}$
12: $\text{cost} \leftarrow$ the value of the cost function (13) with $c_l$ replaced by $b_l$
13: if $\text{cost} < \text{cost}_{\min}$ then
14: $\text{cost}_{\min} \leftarrow \text{cost}$
15: $d_l \leftarrow b_l$, $l = 1, \ldots, L$
16: else
17: $b_{d(k)} \leftarrow 1 - b_{d(k)}$
18: end if
19: end for

6.3. Light Control Flowchart

The system regularly makes observations of the scene but only updates the light setting when user changes positions or activities. The flowchart of the light control system is shown in Fig. 6. When the system detects that the user’s position or activity has changed, it checks the new utility values for all the occupied points (11). If any one of the points has a utility value lower than the pre-defined value $u_{\min}$, the system re-optimizes the light setting.

7. Performance Evaluations

7.1. Activity Analysis

In this section we discuss the performance of the activity analysis methods based on HMM and CRF. The training sequences consisted of 6 video segments
for 3 different users, in which each user walked, sat, and sometimes lay down. Activity was labeled for each frame of each camera. Each segment was about 90 seconds, with a total of $6 \times 3 \times 90 \text{sec} \times 10 \text{frm/sec} = 16200 \text{frm}$ of training samples. During inference, activity was inferred for each camera based on its bounding box aspect ratio and the user height calculated collectively by the network. We observed that lying was more difficult to recognize than sitting, similarly sitting was more difficult to recognize than walking. One reason is that when a user is sitting or lying down, he may be occluded from some views. Another reason is, image features can be view-variant. For example, when a user is lying, a frontal camera may detect a more discriminative bounding box, while others may have bounding boxes similar to those from other activities. Therefore we gave priorities to the “lying” and “sitting” activities, and “lying” has a higher priority than “sitting”. That is, if one camera deducts “lying”, it is selected and the deduction is set to the state of the user.

The performance data is presented in Table 1, where we compare the performance of the HMM and temporal CRF methods in recognizing activities in test sequences with the same amounts of data as the training sequence. There is a noticeable performance improvement for CRF over HMM in recognizing “lying”. Since in the training sequences there are only a few lying instances, with HMM the model is biased by the low frequency of “lying”, and tends to be reluctant to classify an activity as lying. In CRF the activities are learned conditioned on the observations, therefore it has no bias on individual types of activities. It can be observed that the detection results from CRF-based activity analysis are adequate for our control algorithms.
### Table 1: Confusion matrix of activity recognition using HMM and CRF.

<table>
<thead>
<tr>
<th></th>
<th>walking</th>
<th>seated</th>
<th>lying</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>walking</td>
<td>0.900</td>
<td>0.028</td>
<td>0.018</td>
<td>0.054</td>
</tr>
<tr>
<td>seated</td>
<td>0.107</td>
<td>0.808</td>
<td>0.063</td>
<td>0.022</td>
</tr>
<tr>
<td>lying</td>
<td>0.004</td>
<td>0.116</td>
<td>0.794</td>
<td>0.086</td>
</tr>
<tr>
<td>NA</td>
<td>0.006</td>
<td>0</td>
<td>0</td>
<td>0.994</td>
</tr>
</tbody>
</table>

(a) HMM

<table>
<thead>
<tr>
<th></th>
<th>walking</th>
<th>seated</th>
<th>lying</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>walking</td>
<td>0.871</td>
<td>0.117</td>
<td>0</td>
<td>0.011</td>
</tr>
<tr>
<td>seated</td>
<td>0.010</td>
<td>0.903</td>
<td>0.087</td>
<td>0</td>
</tr>
<tr>
<td>lying</td>
<td>0</td>
<td>0.088</td>
<td>0.913</td>
<td>0</td>
</tr>
<tr>
<td>NA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) CRF

### 7.2. Light Control Algorithms

This section discusses the light control simulation results based on occupancy reasoning in our experiments. The experimental data was recorded by 8 cameras. The cameras were calibrated using the method for multi-camera self calibration [24]. The height of the cameras is around 3 meters, and they look toward the center of the room covering most of the area. The layout of the room with cameras is shown in the see Fig. 1(a). Sequences were recorded with a resolution of $320 \times 240$. The size of the room is $9 \times 7 \ m^2$. For the parameters used in the optimization formulation, we set $\delta_1 = 0.3$, $\delta_2 = 0.001$, $I_{\min} = 50$, and $I_{\max} = 800$.

Videos of some recorded simulation results are available online\(^1\). Some snapshots from these results for the intensity control method are shown in Figs. 7, 8, 9, and 10. In each sub-figure, the left upper corner shows the layout of the room, where the 8 cameras are shown by the red squares, and the occupied area is shown by the blue solid ellipsoid. The 8 lights are represented by empty circles while bigger circles imply stronger light intensity. The green dots show the sampled points of the space, brighter color means the utility value of that point is higher. The curves on the bottom of each figure show the intensity of each light over time, and on the right side we show the raw image from one of the cameras.

In Fig. 7, 8, 9, and 10, we show different activities from 4 different users. Fig. 7 illustrate two instances where the first user is walking and sitting. In Fig. 8, the second user is walking and lying, and so on. It can be observed that when a user is sitting, the light is brighter, so the circles representing the lights are bigger or more circles exist. On the other hand, when a user is lying, the circles become smaller compared to other activities. Most of the time, the green dots show bright green, which means that the utility values are high around the

\(^1\)The video can be downloaded at http://wsnl.stanford.edu/videos/occupancy/
Some snapshots from the switch control method are shown in Figs. 11, 12, 13, and 14. In this simulation, we assume that there are 156 lights represented by the small red circles, and these lights can only be turned on or off. In each sub-figure, the curve on the bottom shows the total number of lights that are turned on over time. It can be observed that more lights are turned on when the user is sitting than when he/she is walking, and much fewer lights are turned on when the user is lying. Most of the time, the green dots show bright green, which means that the utility values are high around the area.

To further quantify the performance, we also define the normalized energy efficiency ratio, which is the amount of saved energy from turning off the lights divided by the amount of required energy if all the lights are turned on. Based on the video sequences, we computed the trade-off between the utility values (satisfaction) and the normalized energy efficiency as shown in Fig. 15. The trade-off curves are plotted based on different values of the trade-off parameter \( \delta_1 \) (0.18, 0.22, 0.26, 0.30, 0.34, 0.38, 0.42, 0.46, 0.5), and it shows the average result from all the video sequences. During the simulation, we estimated the utility values of the user and computed the normalized energy efficiency. In general, the system needs to spend more light energy in order to provide more satisfactory light conditions. Therefore, it can be observed that if the energy efficiency is lower, a user has higher satisfaction. But as the curves indicate, we can save more than 75% of the light energy. In practice, based on the application, we can operate the system at the different points on the curve by changing the trade-off parameter.

In general, camera deployment has an impact on the accuracy of the location and position estimation. Issues related to the deployment and calibration of cameras have been investigated, for example, in [9]. In the light control application proposed in our work, we assume that the user is always observed in the area of interest by at least one camera. On the issue of user’s location accuracy, one can argue that since the estimated area from the occupancy-based estimator is always larger than the actual occupied area, in the worst case, for example when only one camera can see the user, the system may turn on more lights than necessary. As for the user’s activity estimation accuracy, the system is capable of classifying the pose by one camera albeit with less accuracy than when multiple cameras collaborate. Therefore, pose estimation accuracy issues would also have a small effect, and in the worst case, when the user’s pose cannot be estimated, the system will default to using the generic utility function.

8. Conclusions

We proposed a user-centric light control algorithm for smart environments. In order to provide suitable light settings for different scenarios, we considered a light control algorithm based on the observations from a camera sensor network. By learning a user’s position and activity, the system can adjust the light setting to provide a more comfortable environment for the user while saving energy by
turning off unnecessary lights. We first discussed the occupancy sensing and human activity analysis methods employed in our system, and then introduced two optimization algorithms for intensity and switching light control. Additionally, in order to save more transmission energy and bandwidth during data gathering, we also proposed the ellipsoid approximation method, which allows the sensors to aggregate (fuse) data during data gathering. The proposed method requires low computation power and can be processed in a distributed fashion across the network nodes. Finally, we presented the light control simulation results based on the occupancy data from multi-camera sequences and presented the trade-off curves of the user satisfaction and the light energy efficiency.

For future work, we plan to improve our human activity analysis algorithm to include a larger variety of activities. Moreover, in this paper, we assumed that the provided light intensity to the user can be calculated, while in our future work we plan to deploy illumination sensors and incorporate a real-time light intensity measurement as a way of feedback in the system. Another area for future work is to allow adaptation of the utility functions for different activities to the preferences of the user using a simple, intuitive interface. This can be accomplished, for example, through a user interface device by which the user provides incremental level changes to the lighting intensity offered by the system for a certain pose. Such feedback can be used by the system to adjust the shape of the corresponding utility function for subsequent instances of observation. A similar extension can be made to allow adaptation with respect to location based on user feedback.

References


Figure 7: Snapshots of the light intensity control results for user 1.
Figure 8: Snapshots of the light intensity control results for user 2.
Figure 9: Snapshots of the light intensity control results for user 3.

(a) User 3 sitting

(b) User 3 lying
Figure 10: Snapshots of the light intensity control results for user 4.
Figure 11: Snapshots of the light switch control results for user 1.
Figure 12: Snapshots of the light switch control results for user 2.
Figure 13: Snapshots of the light switch control results for user 3.
Figure 14: Snapshots of the light switch control results for user 4.

(a) User 4 sitting

(b) User 4 lying
Figure 15: Trade-off curves of the utility values (satisfaction) and the normalized energy efficiency.


