Illumination Adaptation with Rapid-Response Color Sensors

Xinchi Zhang Quan Wang Kim L. Boyer

Signal Analysis and Machine Perception Laboratory
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ABSTRACT

Smart lighting solutions based on imaging sensors such as webcams or time-of-flight sensors suffer from rising privacy concerns. In this work, we use low-cost non-imaging color sensors to measure local luminous flux of different colors in an indoor space. These sensors have much higher data acquisition rate and are much cheaper than many off-the-shelf commercial products. We have developed several applications with these sensors, including illumination feedback control and occupancy-driven lighting.

Keywords: Color sensor, illumination feedback control, occupancy-driven lighting, real-time application

1. INTRODUCTION

Many smart building solutions have been developed based on surveillance video cameras.[1–4] However, the high dimension information provided by video cameras is not necessary for smart lighting control, where identification of individuals and activity recognition are not required. Using video cameras not only makes people feel uncomfortable in a monitored space, but also has a significant privacy concern of information leaking. Non-imaging sensors such as passive infrared (PIR) sensors and ultrasonic sensors have been used as alternatives to cameras.[5,6] In this work, we develop lighting control strategies using distributed low-cost non-imaging color sensors. One color sensor, based on photodiodes and color filters, only generates a few numeric values measuring the local luminous flux of different color channels. Thus we can consider each color sensor as a single-pixel camera. Therefore, using these non-imaging color sensors will not cause any privacy concern. In this paper, we show that such sensors suffice for the task of intelligent control of lighting systems.

2. TESTBED SETUP

We have built a Smart Space Testbed (SST) to implement intelligent lighting systems based on rapid-response color sensors (Figure 1a, 1b). Our testbed is equipped with twelve color controllable LED fixtures, ten color sensors, and a fluorescent lamp for simulating external light sources. All of them are mounted in the ceiling (Figure 1c, 1d). The testbed is 85.5 inches wide, 135.0 inches long, and 86.4 inches high (Figure 1e).

The twelve LED fixtures in the smart space are 7″ LED Downlight Round RGB (Vivia 7DR3-RGB) products from Renaissance Lighting. We can specify the intensity of three color channels of each LED fixture: red, green, and blue. The input to each channel is scaled in the range $[0, 1]$.

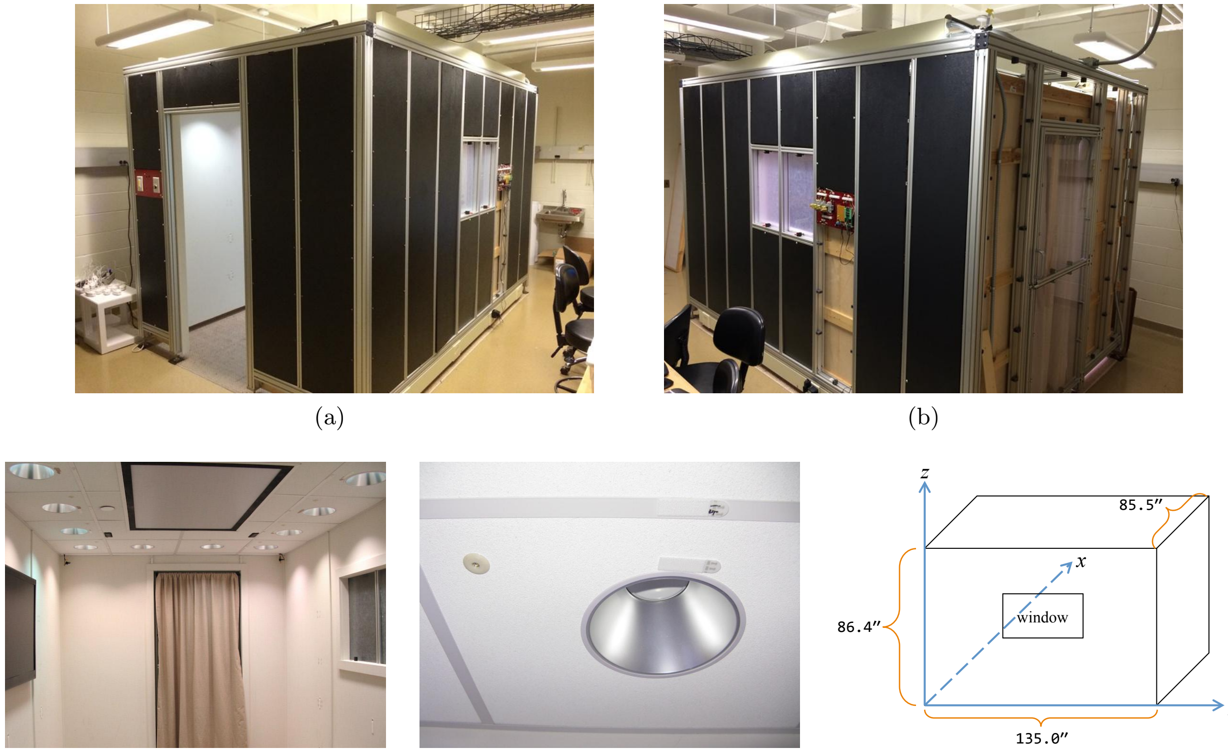
We can read the measurements of four color channels from each color sensor: red, green, blue and white. The output of each channel is an integer in $[0, 43008]$. Detailed information about the sensors will be given in Section 3.

We use the Robot Raconteur software[7] for system communication. With the Robot Raconteur software, we can pull data from color sensors via Ethernet, and send input signals to LED fixtures via Bluetooth.

This testbed has been used for many other research projects, including lighting control,[8] pattern recognition,[9] 3D scene reconstruction,[10] and visual tracking.[11]

Further author information: (Send correspondence to Quan Wang)
Xinchi Zhang: E-mail: zhangx17@rpi.edu, Telephone: 1 518 256 1654
Quan Wang: E-mail: wangq10@rpi.edu, Telephone: 1 518 951 8473
Kim L. Boyer: Email: kim@ecse.rpi.edu

Optics and Photonics for Information Processing VIII, edited by Abdul A. S. Awwal,
Khan M. Iftekharuddin, Mohammad A. Matin, Proc. of SPIE Vol. 9216, 921608
© 2014 SPIE · CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2062105



(a) (b) (c) (d) (e)

Figure 1: The Smart Space Testbed setup. (a) The front view. (b) The rear view. (c) Twelve color controllable LED fixtures and one fluorescent lamp (turned off) illuminate the room from the ceiling. (d) Each color sensor is mounted near an LED fixture. (e) The coordinate system of the room.

3. RAPID-RESPONSE COLOR SENSORS

Although several off-the-shelf color sensing products are available, such as the colorBUG wireless optical light sensor by PIXELTEQ*, these sensors are usually expensive, slow, and not customizable. In our work, we build our own color sensors using easily accessible components.

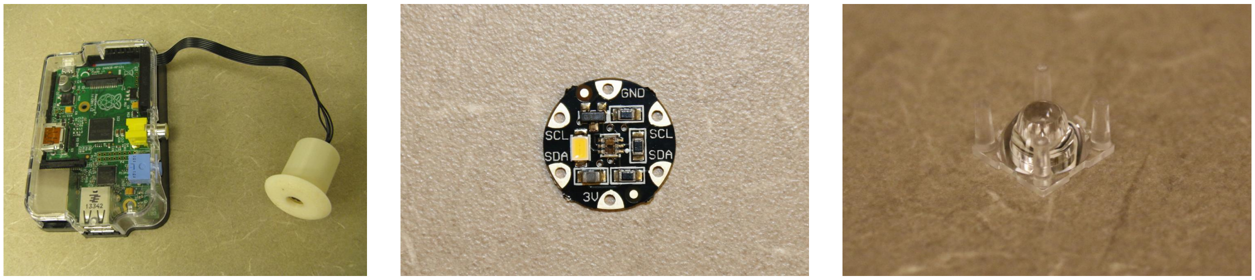
3.1 Components

Our sensors are assembled mostly using commercially available products. Each color sensor is assembled using a Raspberry Pi machine (Model B 512MB RAM), a Flora color sensor chip TCS34725 (Figure 2b), a 3D printed acrylonitrile butadiene styrene (ABS) bushing, a polycarbonate lens (Figure 2c), an SD card, and a general-purpose input/output (GPIO) ribbon cable. We will call our color sensors the *RPi sensors* in the context.

The Raspberry Pi machine is used for GPIO connection and data communication. The operating system and the driver of the Flora color sensor chip (written in Python) are on the SD card. The driver turns off the white illumination LED on the Flora chip after the operating system starts running, and sends the Flora sensor measurements to a computer connected to the Raspberry Pi machine via Ethernet.

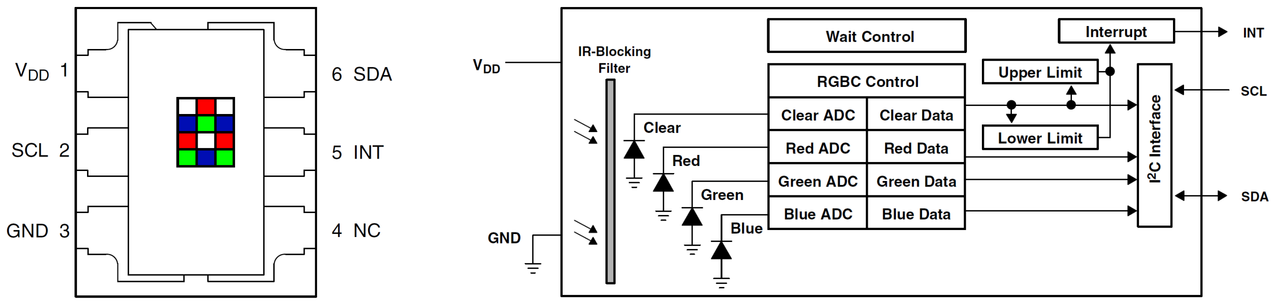
The Flora color sensor chip TCS34725 is a commercial product which can measure local color luminous flux very quickly†. Each TCS34725 chip contains a 3×4 photodiode array: 3 red-filtered photodiodes, 3 green-filtered photodiodes, 3 blue-filtered photodiodes, and 3 clear (unfiltered) photodiodes (Figure 3a). The integration time of the sensor can be customized to any multiples of 2.4 ms no larger than 614.4 ms. In our experiments, we set the integration time to 100.8 ms to strike a balance between accuracy and speed. The Robot Raconteur

*http://www.pixelteq.com/product/colorbug/
†http://www.adafruit.com/datasheets/TCS34725.pdf



(a) (b) (c)

Figure 2: The rapid-response color sensor. (a) Each sensor is based on a Raspberry Pi machine. (b) The flora color sensor chip TCS34725. (c) The polycarbonate lens.



(a) (b)

Figure 3: The TCS34725 chip. (a) The 3×4 photodiode array. (b) The functional block diagram.

software version 0.4 supports reading all color sensors asynchronously, thus using multiple sensors will not cause noticeable delays.

The ABS bushing is designed to hold the lens and the Flora chip. The bushing can be easily mounted in the ceiling. The lens installed in front of the Flora sensor chip is used to transmit more light to the photodiodes.

3.2 Installation

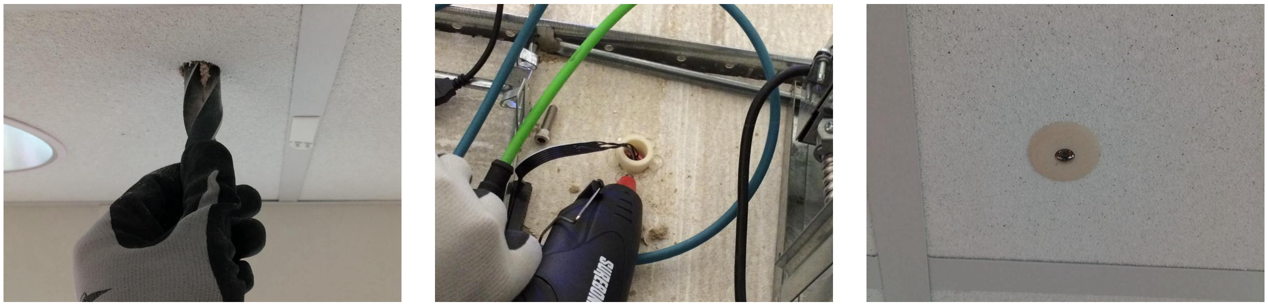
Once all the parts are properly soldered and programmed, we use hot-melt adhesive (also known as hot glue) to stick the bushing, the lens and the Flora chip together (Figure 2a). We use a drill bit to drill holes on the ceiling (Figure 4a), place the bushings through the holes, and stabilize the bushings by applying hot-melt adhesive on the other side of the ceiling (Figure 4b). A 16-port Ethernet switch on the ceiling is used to connect all RPi sensors to a router, which is connected to the computer running Robot Raconteur.

4. REAL-TIME APPLICATIONS

4.1 Illumination Feedback Control

4.1.1 Motivation

With modern color tunable LEDs, in a real space where people live in, in order to improve energy efficiency, we want to maintain the brightness or color temperature of the light field in the space at a specific level. When there are external sources such as sunlight that illuminate the space together with the LED fixtures, we want to adapt the LED input to the changes of external sources, to maintain the light field constant. For example, when strong sunlight illuminates a room from the window, the LED fixtures should become dimmer to reduce energy consumption; when there is no sunlight at night, the LED fixtures should become brighter to enhance human comfort and visibility. Thus, we want to create an illumination feedback control system that adjusts the LED input signals according to the actual light field that is measured by the sensors. Users of this system should be



(a) (b) (c)

Figure 4: The installation of the RPi sensors. (a) A drill bit is used to drill holes on the ceiling. (b) Hot-melt adhesive is used to stabilize the ABS bushings. (c) How the RPi sensors look like from inside the testbed.

able to select a predefined desired light field, and this system will change the input signals to the LED fixtures according to the readings of the color sensors to maintain the light field at the desired level.

We have carefully designed six predefined light field modes: *Dim*, *Normal*, *Bright*, *CT 3000K*, *CT 4500K*, and *CT 6000K*. The first three modes are white light with different intensities; the last three modes are different color temperatures (CT).[12–15]

4.1.2 Controller Design

For the feedback control algorithm, we use a proportional-integral-derivative (PID) controller[16] to control the error — the difference between the desired sensor output and the actual sensor output. And we pair each color sensor with its nearest LED fixture: each color channel of a fixture is controlled only using the corresponding color measurements from its nearest RPi sensor. This practice is also referred to as *decentralized control*.[17,18]

If r is our desired lighting condition, or also called the setpoint (SP), and y is the actual measurement from one sensor, or also called the process variable (PV), then the error is defined as:

$$e = r - y. \tag{1}$$

The PID controller can be written as:

$$\frac{dx}{dt} = K_p \cdot e + K_i \cdot \int_{t-\Delta t}^{t} e\, d\tau + K_d \cdot \frac{de}{dt}, \tag{2}$$

where x is the input of the LED fixture that corresponds to the sensor. K_p, K_i and K_d are the gains of the proportional term, the integral term, and the derivative term, respectively. Δt is the time interval within which we compute the integral term.

The discrete form of Eq. (2) is

$$x_{t+1} = x_t + K_p \cdot e_t + K_i \cdot \sum_{\tau=0}^{T-1} e_{t-\tau} + K_d \cdot (e_t - e_{t-1}), \tag{3}$$

where T corresponds to Δt in Eq. (2). In our experiments, we set $T = 5$.

Figure 5 shows the block diagram of the illumination feedback control system.

The selection of the PID parameters will be discussed in Section 5.1.

4.1.3 User Interface

We have designed a graphical user interface (GUI) for this illumination feedback control system, as shown in Figure 6. Users can specify the light field to one of the six predefined modes using six buttons. The canvas on the left plots the sum of errors of each color channel over all sensors in different control cycles in real time. The pulse in the current canvas is caused by changing the desired level from the “Normal” mode to the “Dim” mode.

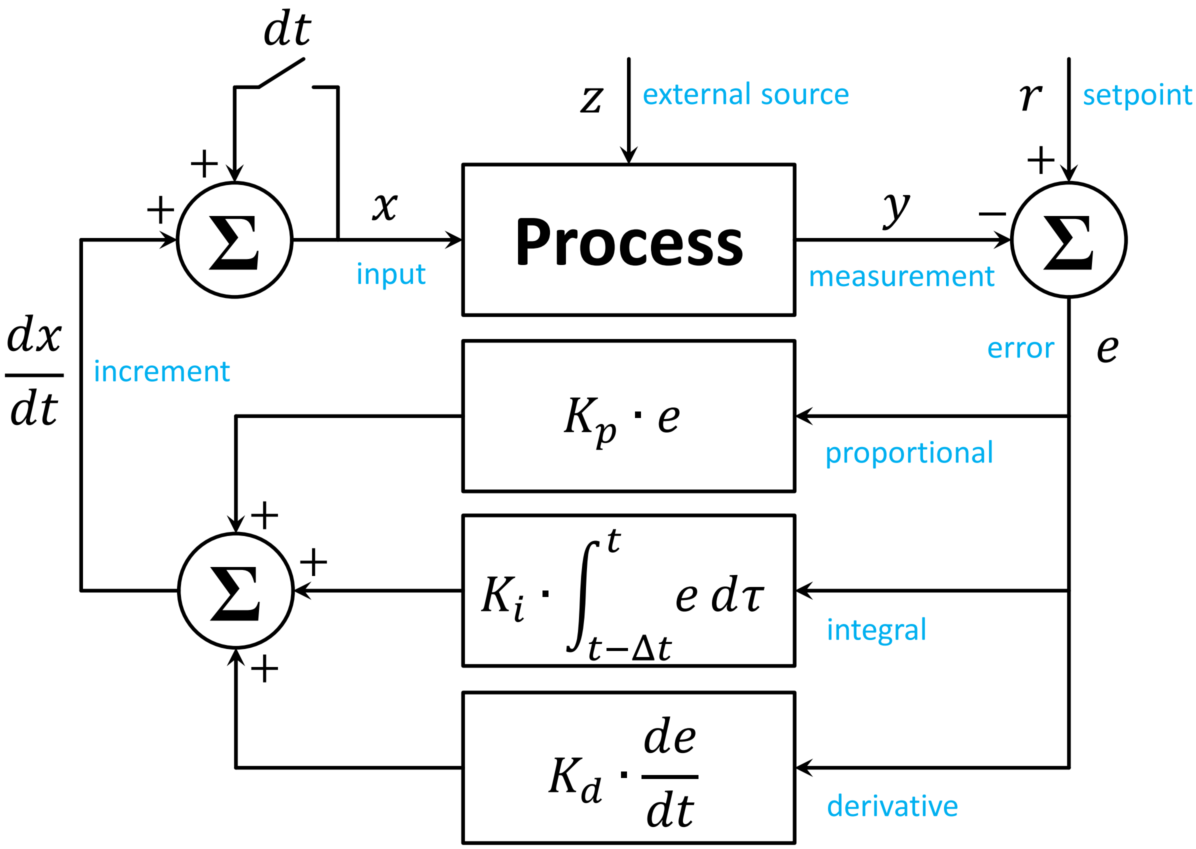


Figure 5: The block diagram of the illumination feedback control system.

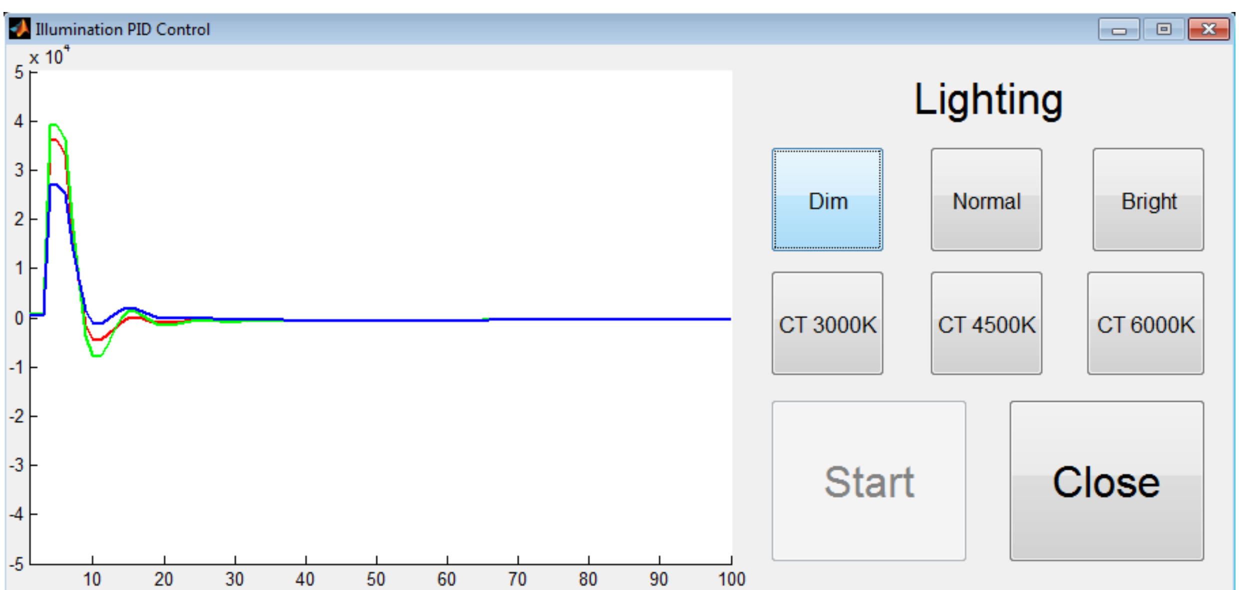


Figure 6: The GUI for the illumination feedback control system.

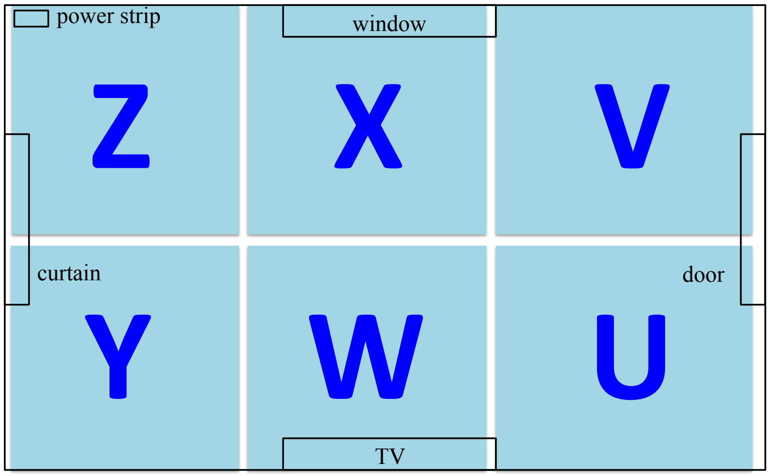


Figure 7: The Smart Space Testbed is divided into 6 regions.

4.2 Occupancy-Driven Lighting

4.2.1 Motivation

To further improve energy efficiency, we also want to deliver light according to the occupancy distribution in the space. For example, if there is only one person in a large room, we may want to only illuminate where this person stands. We call this *occupancy-driven lighting*. We have developed a system which uses machine learning methods to determine which region of the room is being occupied, and delivers more light to that region. Meanwhile, the system dims the fixtures in regions that are not occupied to reduce energy consumption.

4.2.2 Model Training

In our experiments, we divide the room into six regions: U, V, W, X, Y, and Z (Figure 7). Then we classify the occupancy distribution with seven categories: the room is empty or one of the six regions is occupied. We have also designed seven different configurations of LED input signals corresponding to different categories: for the empty room, all LEDs have the same input signals; if one region is occupied, LEDs in that region are brighter, and other LEDs are dimmer. We will refer to these seven LED configurations as *Uniform Light*, *U Brighter*, *V Brighter*, *W Brighter*, *X Brighter*, *Y Brighter*, and *Z Brighter* in the context.

When collecting the training data, for each configuration of LED input signals, we collect the sensor response for scenarios of all seven categories: room being empty, and each region of the room being occupied. 100 measurements are collected for each LED configuration and each category, thus the entire training data contain $7 \times 7 \times 100 = 4900$ measurements. Then we train a support vector machine (SVM)[19] classifier for each LED configuration. Since we repeat this procedure for all seven LED configurations, we actually train seven SVM models, and each SVM model is a 7-category classifier. For the implementation of SVM, we use the well-known LIBSVM library.[20] We also point out that, although SVMs are used as the classifiers in our work, other classifiers such as naive Bayes classifiers[21] or decision trees[22,23] should also work well.

4.2.3 Run Time Logic

With the trained SVM models, at each run time cycle, first the classifier is selected according to the current configuration of LED input. Then this classifier predicts an occupancy label (U, V, W, X, Y, Z, or empty) using the current sensor readings. This predicted label is used to determine which LED configuration should be used in the next cycle. To ensure steady performance, we change the LED configuration only if two consecutive predicted labels are the same. We provide a flowchart in Figure 8 to describe our occupancy-driven lighting system at run time.

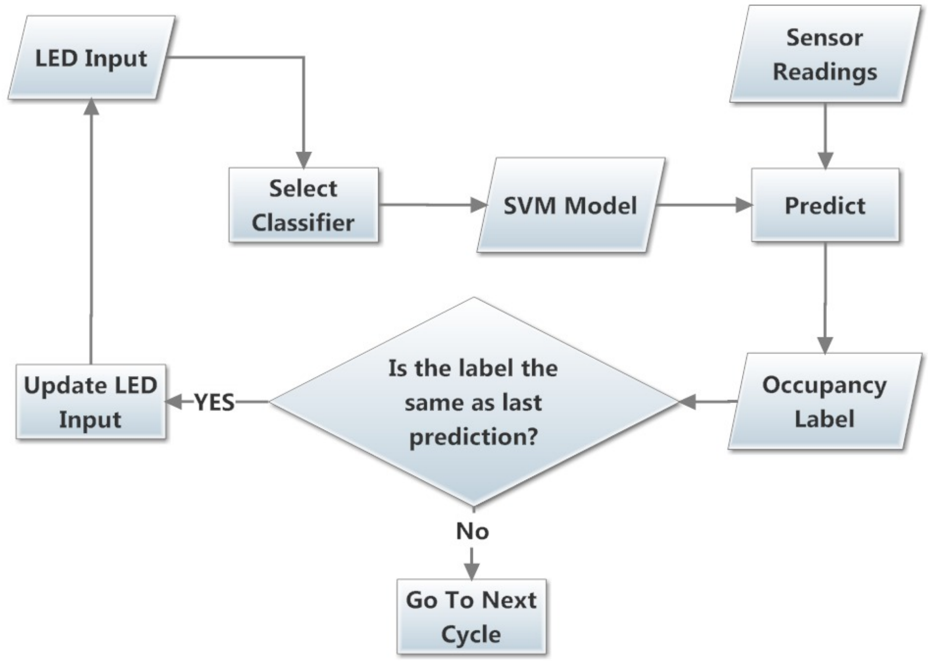


Figure 8: The flowchart for one cycle of the occupancy-driven lighting system at run time.

4.2.4 User Interface

A GUI is created for the occupancy-driven lighting system at run time, as shown in Figure 9. The upper region shows the readings of all RPi sensors. The picture in the bottom left indicates the predicted room occupancy scenario and the current configuration of LED input signals. The text region displays the prediction results and the running time of each cycle.

5. EXPERIMENTAL RESULTS

5.1 Illumination Feedback Control

We have performed experiments to analyze the system response to a step change in the setpoint r. Figure 10 shows the errors of the three color channels of one sensor. The responses of other sensors are very similar. The step change is from the "Dim" mode to the "Bright" mode. Here we use the settling time and the overshoot averaged over all sensors and all channels to evaluate different combinations of PID parameters. For the settling time, we choose 2% as the settling error bound.

First, by rough experiments, we have found a range for the three PID parameters that results in acceptable performance:

$$3 \times 10^{-5} \leq \quad K_p \quad \leq 7 \times 10^{-5}, \tag{4}$$
$$1 \times 10^{-8} \leq \quad K_i \quad \leq 3 \times 10^{-8}, \tag{5}$$
$$2 \times 10^{-5} \leq \quad K_d \quad \leq 6 \times 10^{-5}. \tag{6}$$

Then we have performed brute force search in this range to evaluate different combinations of the three parameters. The performance is provided in Table 1. Since when $K_p = 3 \times 10^{-5}$ or $K_i = 3 \times 10^{-8}$, the system takes too long to settle, we do not report these two cases. From Table 1, we choose the parameter combination with the best performance for our system:

$$K_p = 5 \times 10^{-5}, \tag{7}$$
$$K_i = 1 \times 10^{-8}, \tag{8}$$
$$K_d = 2 \times 10^{-5}. \tag{9}$$

Figure 9: The GUI for the occupancy-driven lighting system at run time.

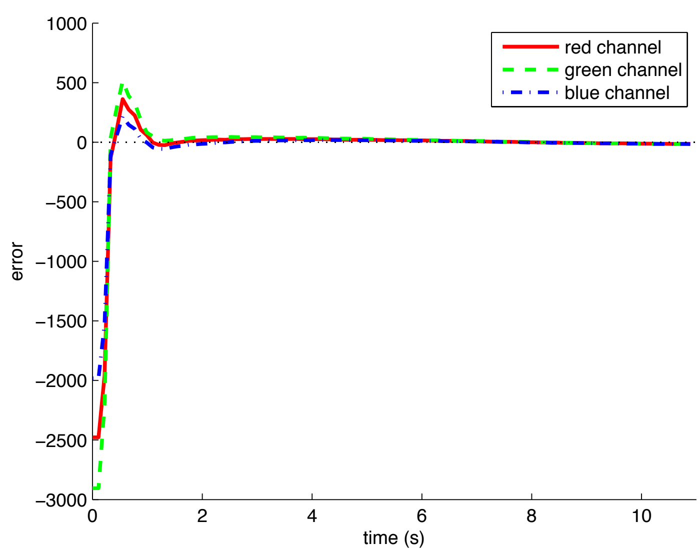


Figure 10: The system response (one sensor) of our PID controller to a step change in the setpoint r applied at time 0.

Table 1: Performance of different PID parameter combinations for the illumination feedback control system.

K_p	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}	5×10^{-5}
K_i	1×10^{-8}	1×10^{-8}	1×10^{-8}	2×10^{-8}	2×10^{-8}	2×10^{-8}
K_d	2×10^{-5}	4×10^{-5}	6×10^{-5}	2×10^{-5}	4×10^{-5}	6×10^{-5}
Settling time (s)	**1.74**	1.82	2.36	3.58	3.79	3.89
Overshoot (%)	**20.24**	20.64	26.59	22.00	20.64	26.97

K_p	7×10^{-5}	7×10^{-5}	7×10^{-5}	7×10^{-5}	7×10^{-5}	7×10^{-5}
K_i	1×10^{-8}	1×10^{-8}	1×10^{-8}	2×10^{-8}	2×10^{-8}	2×10^{-8}
K_d	2×10^{-5}	4×10^{-5}	6×10^{-5}	2×10^{-5}	4×10^{-5}	6×10^{-5}
Settling time (s)	1.99	1.94	2.27	2.40	2.61	3.235
Overshoot (%)	35.40	35.73	36.64	35.04	34.34	38.06

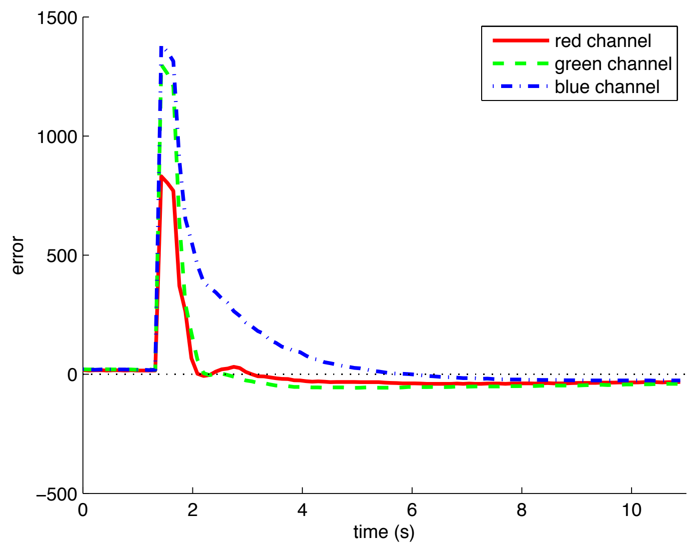


Figure 11: The system response (one sensor) to turning on the fluorescent lamp under the “Bright” mode.

By applying these parameters to Eq. (3), our final PID controller is:

$$x_{t+1} = x_t + 5 \times 10^{-5} \cdot e_t + 1 \times 10^{-8} \cdot \sum_{\tau=0}^{4} e_{t-\tau} + 2 \times 10^{-5} \cdot (e_t - e_{t-1}). \quad (10)$$

We have also performed experiments to evaluate how our system responses to changing external light sources, which can be simulated by using the fluorescent lamp. Figure 11 shows the system response of one sensor to turning on the fluorescent lamp under the “Bright” mode. From Figure 11, we can observe that the sensors were able to sense the color changes caused by the fluorescent lamp, and the system dimmed all fixtures to compensate this change. Since the fluorescent light contains more blue component, it takes longer for the blue channel to settle. The average settling time of all sensors and all channels is 4.11 seconds.

A video demo showing our illumination feedback control system running in real time can be found at http://youtu.be/APXWaWQdWZU.

5.2 Occupancy-Driven Lighting

We have performed cross validation experiments using the collected data. We evaluated both linear SVM and radial basis function (RBF) kernel SVM with different soft margin costs. For the cross validation, we randomly partition the data into training set (67%) and testing set (33%), repeat the process 20 times, and record the

Table 2: Cross validation classification accuracy on the collected data using linear and RBF kernel SVM.

Linear SVM Accuracy (%)						
Soft Margin Cost	0.0001	0.001	0.01	0.1	1	10
Uniform Light	15.13	97.55	99.72	**100.0**	**100.0**	99.83
U Brighter	13.22	96.57	**100.0**	**100.0**	**100.0**	**100.0**
V Brighter	13.82	96.12	99.98	**100.0**	**100.0**	**100.0**
W Brighter	13.09	98.90	99.94	**100.0**	**100.0**	**100.0**
X Brighter	17.02	97.40	99.79	**99.91**	99.81	99.81
Y Brighter	14.42	95.92	99.68	**99.87**	99.81	99.85
Z Brighter	15.19	93.50	**100.0**	**100.0**	99.91	99.98

RBF Kernel SVM Accuracy (%)						
Soft Margin Cost	0.0001	0.001	0.01	0.1	1	10
Uniform Light	13.61	13.24	18.37	99.55	99.94	**100.0**
U Brighter	12.66	12.42	15.69	99.96	**100.0**	**100.0**
V Brighter	13.35	11.42	18.41	**100.0**	**100.0**	**100.0**
W Brighter	13.45	13.03	20.36	**100.0**	**100.0**	**100.0**
X Brighter	13.71	13.20	20.06	99.59	**100.0**	99.98
Y Brighter	14.25	14.29	17.34	99.64	99.64	**99.85**
Z Brighter	13.26	12.81	17.38	99.96	**100.0**	99.98

average classification accuracy on the testing set. Since there are seven LED configurations, the cross validation is carried out for each configuration separately. The results are provided in Table 2. From this table, we can observe that the performance is better with larger soft margin cost, which means that less misclassification is more important than larger margin. For the real-time system, we use RBF kernel SVM with soft margin cost $c = 1$. With the control strategy shown in Figure 8, the system has very accurate and robust performance at run time.

A video demo showing our occupancy-driven lighting system running in real time can be found at `http://youtu.be/ijJmI35yihc`.

6. DISCUSSION: LIMITATIONS AND FUTURE IMPROVEMENTS

6.1 Illumination Feedback Control

In the illumination feedback control application, we paired each sensor with one LED fixture as a decentralized control system. This is because each LED fixture has more impact on the color sensor nearest to it than on other color sensors. However, it still has impact on those non-nearest sensors, and this impact cannot be ignored when the desired brightness level is low. Neglecting such impact will not only result in long settling time, but also result in some uncomfortable LED configurations occasionally. Thus, adjusting each LED fixture only using its nearest sensor is not the best practice. To improve, we can modify the controller to also consider a few non-nearest sensors, and we can also penalize big differences in the input to neighboring fixtures as a post-process.

We also noticed that the response in the blue channel of our sensors is usually less likely to oscillate. We have performed experiments to show that the blue channel is less sensitive than the other two channels. Thus, the PID parameters could be tuned separately for the three different color channels for better performance, instead of using shared parameters for all channels.

We also point out that in the future, the color sensor can be built into the LED fixture circuit as a combined product. This will not only ensure more accurate measurements, but also make it easier to control the total cost of the lighting system.

6.2 Occupancy-Driven Lighting

In the occupancy-driven lighting application, the training and prediction are both based on single measurements. Thus multiple classifiers have to be trained to cover all LED configurations, which is tedious human labor. Also, if there is a changing external light source, the classifiers will fail.

This problem can be addressed by using the light transport matrix as a signature of the occupancy.[10] However, to accurately recover the light transport matrix, multiple measurements are required, and the delay of our current fixtures is the bottleneck of doing this in real time. Using faster LED fixtures will easily solve this problem.

7. CONCLUSION

In this paper, we have presented two smart lighting applications using our rapid-response color sensors: illumination feedback control and occupancy-driven lighting. For the illumination feedback control application, we use a PID controller to control the LED fixtures in a room to maintain the light field at a user specified level. For the occupancy-driven lighting application, we train SVM classifiers using sensor output data collected under different LED configurations, and in real time change the lighting condition according to the occupancy scenario predicted by the classifiers. The biggest challenges of these applications are: (1) To preserve the privacy of users, the color sensors based on photodiodes and color filters contain much less information than other sensors such as cameras. (2) To implement robust real-time applications, the sensors need to response very quickly and the measurements must be accurate. We have carried out experiments to evaluate our two applications, and both applications have satisfying performance.

ACKNOWLEDGMENTS

This work was supported primarily by the Engineering Research Centers Program (ERC) of the National Science Foundation under NSF Cooperative Agreement No. EEC-0812056 and in part by New York State under NYSTAR contract C090145.

The RPi sensors are built collaboratively with Mr. Charles Goodwin and Mr. Lawrence Fan. In the illumination feedback control application, the six predefined light field modes are designed with the help of Mr. Anqing Liu and Mr. Sina Afshari. Besides, the authors would like to thank Prof. Richard J. Radke and Prof. Robert Karlicek for the very helpful discussions.

REFERENCES

[1] Glennie, W., Thukral, I., and Rea, M., "Lighting control: feasibility demonstration of a new type of system," *Lighting Research and Technology* **24**(4), 235–242 (1992).

[2] Kamthe, A., Jiang, L., Dudys, M., and Cerpa, A., "Scopes: Smart cameras object position estimation system," in [*Wireless Sensor Networks*], Roedig, U. and Sreenan, C., eds., *Lecture Notes in Computer Science* **5432**, 279–295, Springer Berlin Heidelberg (2009).

[3] Erickson, V. L., Lin, Y., Kamthe, A., Brahme, R., Surana, A., Cerpa, A. E., Sohn, M. D., and Narayanan, S., "Energy efficient building environment control strategies using real-time occupancy measurements," in [*Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*], 19–24, ACM (2009).

[4] Erickson, V. L., Carreira-Perpiñán, M. Á., and Cerpa, A. E., "Observe: Occupancy-based system for efficient reduction of hvac energy," in [*10th International Conference on Information Processing in Sensor Networks (IPSN)*], 258–269, IEEE (2011).

[5] Guo, X., Tiller, D., Henze, G., and Waters, C., "The performance of occupancy-based lighting control systems: A review," *Lighting Research and Technology* **42**(4), 415–431 (2010).

[6] Hassan, M. Y., Abdullah, H., Rahman, H. A., Abdullah, M. P., Hussin, F., Said, D. M., et al., "A review on lighting control technologies in commercial buildings, their performance and affecting factors," *Renewable and Sustainable Energy Reviews* **33**, 268–279 (2014).

[7] Wason, J. D. and Wen, J. T., "Robot raconteur: A communication architecture and library for robotic and automation systems," in [*IEEE Conference on Automation Science and Engineering (CASE)*], 761–766, IEEE (2011).

[8] Afshari, S., Mishra, S., Julius, A., Lizarralde, F., and Wen, J. T., "Modeling and feedback control of color-tunable led lighting systems," in [*American Control Conference (ACC)*], 3663–3668, IEEE (2012).

[9] Wang, Q., Zhang, X., and Boyer, K. L., "3d scene estimation with perturbation-modulated light and distributed sensors," in [*10th IEEE Workshop on Perception Beyond the Visible Spectrum (PBVS)*], (2014).

[10] Wang, Q., Zhang, X., Wang, M., and Boyer, K. L., "Learning room occupancy patterns from sparsely recovered light transport models," in [*22nd International Conference on Pattern Recognition (ICPR)*], (2014).

[11] Jia, L. and Radke, R., "Using time-of-flight measurements for privacy-preserving tracking in a smart room," *IEEE Transactions on Industrial Informatics* **10**, 689–696 (Feb 2014).

[12] Smith, W. J., [*Modern optical engineering*], Tata McGraw-Hill Education (1966).

[13] Wyszecki, G. and Stiles, W. S., [*Color science*], vol. 8, Wiley New York (1982).

[14] Taylor, A. E. et al., "Illumination fundamentals," tech. rep., Lighting Research Center, Rensselaer Polytechnic Institute (2000).

[15] Borbély, Á., Sámson, Á., and Schanda, J., "The concept of correlated colour temperature revisited," *Color Research & Application* **26**(6), 450–457 (2001).

[16] Ang, K. H., Chong, G., and Li, Y., "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology* **13**(4), 559–576 (2005).

[17] Siljak, D. D., [*Decentralized control of complex systems*], Courier Dover Publications (2011).

[18] Sandell, N. R., Varaiya, P., Athans, M., and Safonov, M. G., "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control* **23**(2), 108–128 (1978).

[19] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine learning* **20**(3), 273–297 (1995).

[20] Chang, C.-C. and Lin, C.-J., "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (ACM TIST)* **2**, 27:1–27:27 (May 2011).

[21] Rennie, J. D., Shih, L., Teevan, J., Karger, D. R., et al., "Tackling the poor assumptions of naive bayes text classifiers," in [*International Conference on Machine Learning (ICML)*], **3**, 616–623, Washington DC (2003).

[22] Wang, Q., Ou, Y., Julius, A., Boyer, K., and Kim, M. J., "Tracking tetrahymena pyriformis cells using decision trees," in [*21st International Conference on Pattern Recognition (ICPR)*], 1843–1847 (Nov 2012).

[23] Wang, Q., Wu, D., Lu, L., Liu, M., Boyer, K. L., and Zhou, S. K., "Semantic context forests for learning-based knee cartilage segmentation in 3d mr images," in [*Medical Computer Vision. Large Data in Medical Imaging*], Menze, B., Langs, G., Montillo, A., Kelm, M., Mller, H., and Tu, Z., eds., *Lecture Notes in Computer Science*, 105–115, Springer International Publishing (2014).