Learning Room Occupancy Patterns from Sparsely Recovered Light Transport Models

Quan Wang Xinchi Zhang Meng Wang Kim L. Boyer
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180, USA
wangq10@rpi.edu zhangx17@rpi.edu wangm7@rpi.edu kim@ecse.rpi.edu

***Abstract*—In traditional vision systems, high level information is usually inferred from images or videos captured by cameras, or depth images captured by depth sensors. These images, whether gray-level, RGB, or depth, have a human-readable 2D structure which describes the spatial distribution of the scene. In this paper, we explore the possibility to use distributed color sensors to infer high level information, such as room occupancy. Unlike a camera, the output of a color sensor has only a few variables. However, if the light in the room is color controllable, we can use the outputs of multiple color sensors under different lighting conditions to recover the light transport model (LTM) in the room. While the room occupancy changes, the LTM also changes accordingly, and we can use machine learning to establish the mapping from LTM to room occupancy.**

***Keywords*—room occupancy; light transport model; color sensors; controllable light**

I. Introduction

Although surveillance video cameras have widely been used for automatic monitoring of indoor spaces, one major concern on this technique is the privacy of human subjects. Staying in a monitored room simply makes people feel uncomfortable. In this work, we seek to infer the room occupancy by using distributed color sensors, where each color sensor can be thought of as a single pixel with no spatial information. Once we are able to determine the room occupancy, we can use such information to perform intelligent control of the room. We provide a flowchart to illustrate our system pipeline in Figure 1.

A. Smart Room Setup

Our testbed is a smart lighting room with one window and two doors, where color controllable LED light fixtures and color sensors are installed. The inner space of the smart room is about 7.12 feet wide, 11.25 feet long, and 7.20 feet high.

Twelve color controllable LED fixtures illuminate the room from the ceiling (Figure 2b). For each LED fixture, we can control the intensity of three color channels: red, green and blue. The input to each channel is scaled to lie in the range $[0, 1]$. We also place twelve Seachanger wireless Colorbug sensors (Figure 2a) on the walls of this room (Figure 2c). Each color sensor has four output channels: red, green, blue and white. We use the Robot Raconteur software [1] for communication. The software connects to color sensors with Wi-Fi, and sends input signals to LED fixtures via Bluetooth. This same smart room has been used for a number of other research projects, including lighting control algorithms [2], [3] and visual tracking systems [4].

II. Light Transport Model

Since the current configuration of our testbed has twelve LED fixtures with three channels each, the input to the system is an $m_1 = 36$ dimensional signal x. Because we have twelve color sensors, each with four channels, the measurement is an $m_2 = 48$ dimensional signal y. We have performed experiments to show that there is an affine relationship between x and y:

$$y = Ax + b, \tag{1}$$

where $A \in \mathbb{R}^{m_2 \times m_1}$ and $b \in \mathbb{R}^{m_2}$. The matrix A is called the light transport matrix, and the vector b is the sensor output with respect to the ambient light. When we set the input to a given level x_0, the output of the sensors is $y_0 = Ax_0 + b$. Now if we add a small perturbation δx to the input, the new output becomes $y_0 + \delta y = A(x_0 + \delta x) + b$. By simple subtraction, we can cancel out b, and get

$$\delta y = A\delta x, \tag{2}$$

which is equivalent to the linear *light transport model (LTM)* described in [5]. Here we call x_0 the *base light*, which is often specified by a control algorithm.

If we measure y_0 once, and measure $y_0 + \delta y$ many times with different δx, then we get a linear system to solve for A. We define the magnitude of δx as $\rho = ||\delta x||_\infty$. The choice of the magnitude is a trade-off: we need it to be large enough to be accurately sensed by the color sensors, but small enough to keep the human subjects feel comforatable. In our experiments, we set $\rho = 0.025$.

III. Model for an Empty Room

When the smart room is empty, we assume the light transport matrix is A_0. We perturb the input of the LED fixtures x_0 with randomly generated m_1-dimensional signals $\delta x_1, \delta x_2, \ldots, \delta x_n$, and measure the m_2-dimensional changes of the sensor readings $\delta y_1, \delta y_2, \ldots, \delta y_n$. Let $X = [\delta x_1, \delta x_2, \ldots, \delta x_n]$ and $Y = [\delta y_1, \delta y_2, \ldots, \delta y_n]$, where $X \in \mathbb{R}^{m_1 \times n}$ and $Y \in \mathbb{R}^{m_2 \times n}$. Now the linear system becomes $Y = A_0 X$. If $n \geq m_1$, this linear system can be solved by the pseudo-inverse:

$$A_0 = YX^\mathsf{T}(XX^\mathsf{T})^{-1}, \tag{3}$$

which corresponds to the minimization of the Frobenius norm of the error:

$$\min_{A_0} ||Y - A_0 X||_F. \tag{4}$$

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.347

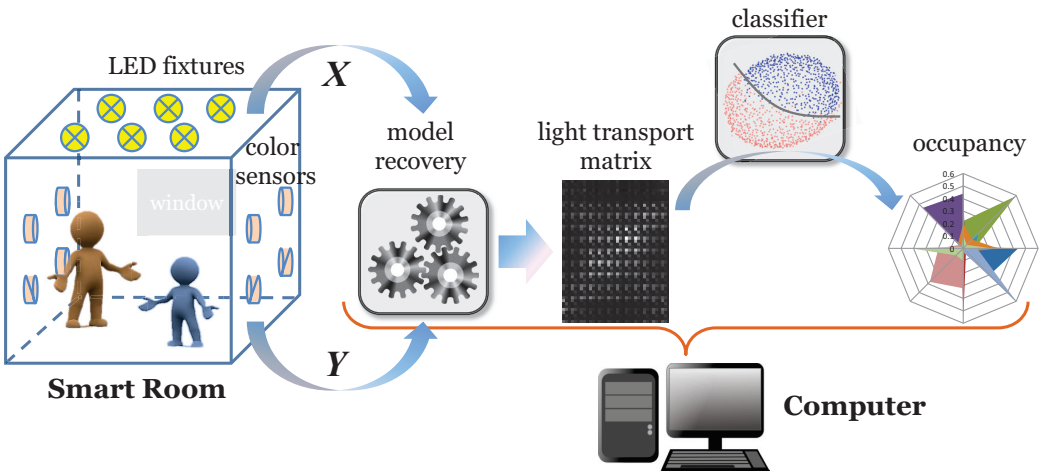

Fig. 1: The flowchart of our room occupancy determination system.

(a) (b) (c)

Fig. 2: The smart room setup. (a) The colorbug sensor. (b) Twelve color controllable LED fixtures illuminate the room from the ceiling. (c) Colorbug sensors are placed on the walls.

Since we are modeling an empty room, this process can be performed offline as a calibration step, and we can make many measurements to ensure $n \geq m_1$.

IV. Sparse Recovery of Light Transport Changes

Suppose we have determined the light transport matrix A_0 for an empty room. Now, if the room occupancy changes, the light transport matrix A will also change. To recover the new A, we again randomly perturb LED input with $X = [\delta x_1, \delta x_2, \ldots, \delta x_n]$, and measure the changes of color sensor outputs $Y = [\delta y_1, \delta y_2, \ldots, \delta y_n]$. We now have the constraints $Y = AX$. To ensure real time performance, we can only take a limited number of measurements within a short time interval, during which we can assume the occupancy, the base light x_0, and the sensor response b to the ambient light are all constant. Thus $n < m_1$ and this is an underdetermined system. To solve for A, we need to add extra constraints. One intuition is that the changes in the light transport model are sparse — when the occupancy changes, most light paths are not affected.

To better understand this philosophy, we need to take an in-depth look into the light transport matrix A. Each entry of A can be thought of as a summation of responses from one sensor channel to one LED channel on all the light paths from this LED to this sensor. These light paths include the line segment connecting the LED fixture and the sensor (direct path), and numerous reflection paths (mostly diffuse reflection). Apparently, the direct path is the dominating component (Figure 3a). When a human subject enters the room, he only blocks a limited number of these direct paths, as illustrated in Figure 3b. Our sparsity assumption can be understood in this way: because the room is sparsely occupied, only a few direct paths will change when the room occupancy changes.

Thus we hope to minimize the difference between A and A_0. Since $Y = AX$, we have $A_0X - Y = (A_0 - A)X$. Let $Z = A_0X - Y$ and $E = A_0 - A$. Z can be interpreted as the change in sensor responses due to the change in room occupancy; E can be interpreted as the change in the light transport model due to the change in room occupancy. The matrix E is referred to as the *light transport changes*. Then our problem becomes $Z = EX$, where we hope E approximates the zero matrix, which can be formulated as:

$$\begin{aligned} \min_{E} \quad & f(E) \\ \text{s.t.} \quad & Z = EX, \end{aligned} \tag{5}$$

where $f(\cdot)$ is a target function describing how deficient or

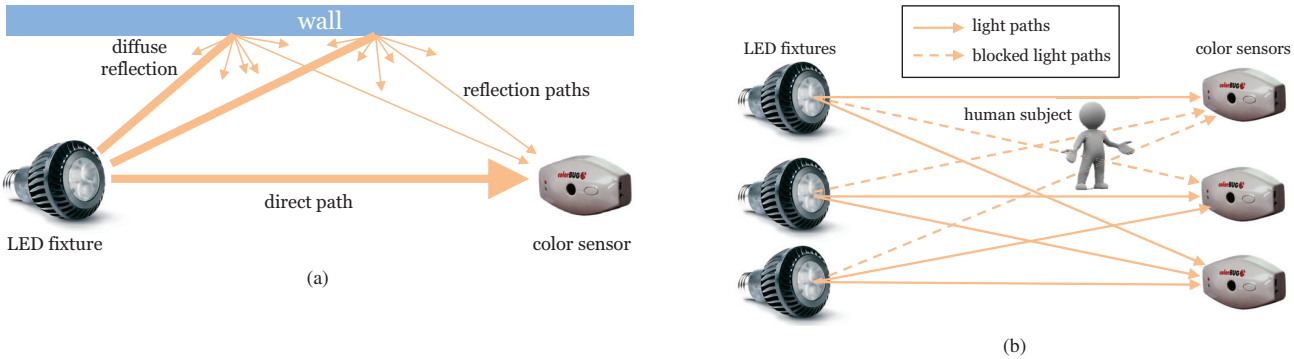

Fig. 3: (a) Direct path is the dominating component. (b) Room is sparsely occupied.

sparse the matrix E is. Here we clarify that E is not really almost zero, because it is the change in the model which should be noticeable when the occupancy changes. We are just looking for the E that is closest to zero matrix under the constraint $Z = EX$. We have different options for the function $f(\cdot)$:

$$f_1(E) = \text{rank}(E), \quad (6)$$
$$f_2(E) = ||E||_F = ||\text{vec}(E)||_2, \quad (7)$$
$$f_3(E) = ||\text{vec}(E)||_0, \quad (8)$$
$$f_4(E) = ||\text{vec}(E)||_1. \quad (9)$$

Here $\text{vec}(\cdot)$ denotes the vectorization of a matrix (stacking the columns into a single column vector). Different choices of the target function $f(\cdot)$ have different practical meanings, and correspond to different assumptions. If we minimize the rank of E, we are assuming that many rows of E are linearly dependent. In another word, the change in the occupancy affect the light paths from the sources to different sensors in a similar way. Minimizing the Frobenius norm of E is not straightforward, but we will later show that its solution is also the solution to rank minimization. Minimizing target function $f_3(E)$ or $f_4(E)$ will result in a sparse matrix E, where many entries of E are zeros. A sparse E means that only a limited number of direct light paths have changed as a result of the change in the room occupancy.

A. Rank Minimization

The rank minimization problem we propose here is different from the matrix completion problem [6], where people seek to recover a low-rank matrix from a small subset of its entries. Our constraint is a linear system, not a known subset of entries. To solve the problem in Eq. (5) with target function (6), let the singular value decomposition (SVD) of X be $X = USV^\mathsf{T}$, where $U \in \mathbb{R}^{m_1 \times m_1}$, $S \in \mathbb{R}^{m_1 \times n}$, and $V \in \mathbb{R}^{n \times n}$. Then the constraint becomes $ZV = EUS$. Since $n < m_1$, we can write S as $S = \begin{bmatrix} S_1 \\ 0 \end{bmatrix}$, where S_1 is the $n \times n$ diagonal matrix of the singular values of X. Let $EU = F = [\ F_1 \quad F_2\ ]$, where F_1 is $m_1 \times n$ and F_2 is $m_1 \times (m_1 - n)$. Now our constraint is simply $F_1 S_1 = ZV$, or $F_1 = ZVS_1^{-1}$, and F_2 can be an arbitrary submatrix. Since the rank is invariant under unitary transforms, we know

$$\text{rank}(E) = \text{rank}(EU) = \text{rank}(F) \geq \text{rank}(F_1). \quad (10)$$

Apparently, when $F_2 = 0$, $\text{rank}(E) = \text{rank}(F_1)$, thus our solution is

$$E = [\ ZVS_1^{-1} \quad 0\ ]\, U^\mathsf{T}. \quad (11)$$

An interesting observation here is that our solution to the rank minimization problem is also a solution to the Frobenius norm minimization problem. This is because the Frobenius norm is also invariant under unitary transforms:

$$||E||_F^2 = ||EU||_F^2 = ||F||_F^2 = ||F_1||_F^2 + ||F_2||_F^2, \quad (12)$$

and it takes its minimum also when $F_2 = 0$.

We also point out that the rank minimization problem can also be viewed as a sparse recovery problem, because the rank of a matrix is simply the number of non-zero singular values of this matrix [7]. Thus minimizing the rank of a matrix is equivalent to minimizing the ℓ_0 norm of its vector of singular values.

B. Sparse Recovery

To solve the problem in Eq. (5) with target function (8) or (9), we can rewrite $Z = EX$ to its Kronecker product form:

$$\text{vec}(Z) = (X^\mathsf{T} \otimes I_{m_2 \times m_2})\, \text{vec}(E). \quad (13)$$

This is a standard compressed sensing problem, where $\text{vec}(E)$ is the unknown sparse signal of interest, $\text{vec}(Z)$ is the vector of measurements, and the Kronecker product $\Phi = X^\mathsf{T} \otimes I_{m_2 \times m_2}$ is the sensing matrix.

1) Solutions: Minimizing $f_3(E)$ is an ℓ_0 optimization problem, which is NP-hard. However, greedy algorithms such as orthogonal matching pursuit (OMP) [8] can be used to find the approximate solution efficiently. We use the SparseLab toolbox [9] to solve the ℓ_0 optimization problem. Minimizing $f_4(E)$ is an ℓ_1 optimization problem, which can be recast as a linear programming problem [10]. We solve for the ℓ_1 optimization problem using the ℓ_1-magic toolbox [11].

2) Theoretical Guarantee of ℓ_1 Minimization: By minimizing $f_4(E)$, we wish to find the sparse signal $\text{vec}(E)$. However, ℓ_1 minimization can only recover the sparse signals under certain conditions. The isometry constant [12] of order s of

a sensing matrix Φ is defined as the smallest number δ_s such that

$$(1-\delta_s)||x||_2^2 \leq ||\Phi x||_2^2 \leq (1+\delta_s)||x||_2^2 \quad (14)$$

holds for all s-sparse signals x (an s-sparse signal is a signal where only s elements are non-zero). According to the noiseless recovery theorem in [10], if $\mathrm{vec}(E)$ is s-sparse and the isometry constant of order $2s$ of the sensing matrix Φ is δ_{2s}, then a sufficient condition for ℓ_1 minimization to exactly recover $\mathrm{vec}(E)$ is $\delta_{2s} < \sqrt{2}-1$.

There are existing results on the restricted isometry property (RIP) analysis of matrix Kronecker products [13], [14]. One essential conclusion from the previous work is:

$$\begin{aligned}\max\{\delta_s(A), \delta_s(B)\} &\leq \delta_s(A \otimes B) \\ &= \delta_s(B \otimes A) \\ &\leq (1+\delta_s(A))(1+\delta_s(B)) - 1, \quad (15)\end{aligned}$$

where A and B are two matrices, and $\delta_s(\cdot)$ is the isometry constant of order s of a matrix. It can be easily shown that $\delta_s(I_{m_2 \times m_2}) = 0$ for all s. Thus from Eq. (15) we get:

$$\begin{aligned}\max\{\delta_s(X^\mathsf{T}), 0\} \leq \delta_s(\Phi) &= \delta_s(X^\mathsf{T} \otimes I_{m_2 \times m_2}) \\ &\leq (1+\delta_s(X^\mathsf{T})) - 1, \quad (16)\end{aligned}$$

or simply $\delta_s(\Phi) = \delta_s(X^\mathsf{T})$.

V. Occupancy Determination

The light transport matrix A is dependent only on the room occupancy, and independent of the input to the LED light fixtures. Thus once we are able to recover the matrix A, we can infer high level information about the room occupancy from this matrix. We can either directly use the entries of matrix A as features, or build statistical models for feature extraction. Then we can use these features for supervised learning to discriminate among various occupancy scenarios in the smart room.

A. Classification Problem Setup

We set up two classification problems. The first one is a four-category classification problem, where we want to determine whether the room is empty, occupied by a small group (one to three) of people, occupied by a large group (four to seven) of people gathering in the room, or occupied by a large group of people scattered in the room. In the following sections such as Figure 5, these categories are referred to as "empty", "small", "LG" (large gathered), and "LS" (large scattered), respectively.

The second problem, which is much more difficult, is a fifteen-category classification problem, where we want to discriminate between single individuals and small groups (two to three) of people, and also locate which part of the room they occupy. We manually divide the room into six regions, named "U", "V", "W", "X", "Y", and "Z" (Figure 4). In the following sections such as Figure 6 and Figure 7, the categories of a single individual are named as "1+region", and the categories of a small group (two to three people) are named as "S+region". So the fifteen categories are: empty, six regions for individual, six regions for small group, large group gathered, and large group scattered.

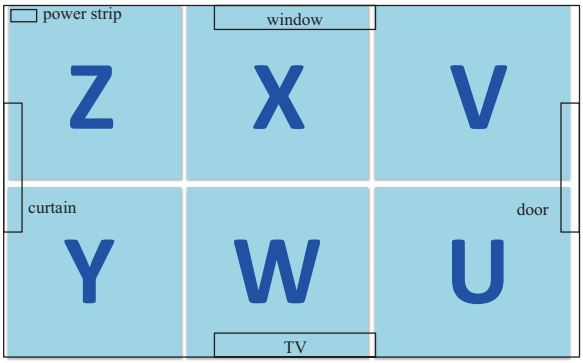

Fig. 4: A diagram of the six regions of the room.

B. Data Collection

We collect data by creating various room occupancy scenarios, randomly generating perturbed inputs to the LED fixtures, and measure the outputs of the Colorbug sensors. To ensure this method can run in real time, we limit our number of measurements to $n = 24$. Since it takes about 0.4 second for our system to send inputs to LED fixtures and read outputs from the Colorbug sensors, the n measurements plus the base light measurement are made within 10 seconds, during which we assume the room occupancy scenario does not change. The occupancy scenarios are created by using nine subjects. Each time we ask several subjects to enter the room and we read Colorbug sensors using different inputs to the LEDs. We record the number and location of the subjects to create the ground truth labels for the two classification problems. Then we randomly partition each class, using 60% as training data and 40% as testing data.

VI. Experimental Results

In this section, we directly use the entries of the matrix E as our features. Thus our feature dimension is $m_1 m_2 = 1728$. These features are then normalized — each entry is subtracted by its mean and divided by its standard deviation over the training set. For each of the classification problems that we have set up, we use radial basis function (RBF) kernel support vector machine (SVM) as the classifier, and use the mean average precision (mAP) as our goodness measurement. Because different categories have different sizes, other performance measures such as accuracy can be highly biased, especially when some easiest classes are significantly larger than others. For each category, we train a one-versus-all SVM, and use the decision scores to compute the average precision (AP). Then the mAP is simply the mean value of the average precisions for all categories.

A. Four-Category Classification

Using rank minimization to recover light transport changes, we are able to achieve a mAP of 88.62%, compared with the mAP of random guess being 25.54%. Using ℓ_1 minimization, we achieve a mAP of 81.92%. Using ℓ_0 minimization, the mAP is 71.19%. The average precisions of each category using different methods are shown in Figure 5.

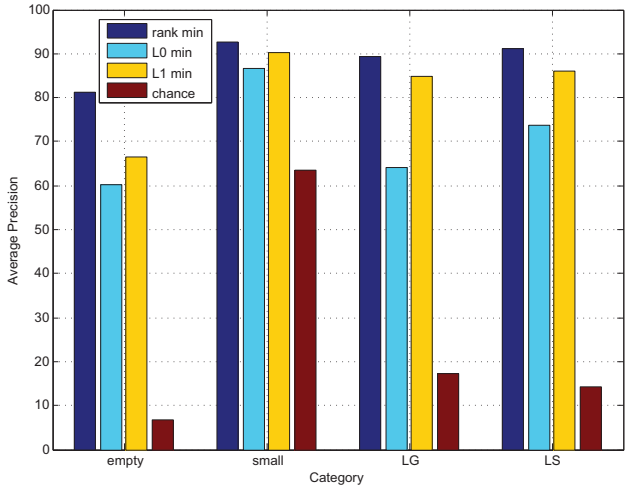

Fig. 5: The average precisions (AP) for the four-category classification problem. The four categories are: empty room, small group (one to three people), large group (four to seven people) gathered, and large group scattered.

In Figure 5, the AP of the second category using random guess is much higher than other categories because this category has a much larger size. For the four-category classification problem, we can see the light transport changes recovered by rank minimization method produces the best classification performance. This means that our assumption that the change in the occupancy affect the light paths from the sources to different sensors in a similar way is reasonable.

B. *Fifteen-Category Classification*

For the fifteen-category classification problem, using rank minimization to recover light transport changes, we are able to achieve a mAP of 78.69%, compared with the mAP of random guess being 7.95%. Using ℓ_1 minimization the mAP is 68.37%, while using ℓ_0 minimization the mAP is 50.76%. The average precisions of each category using different methods are shown in Figure 6.

As expected, the performance of this problem is worse than only four categories. However, the mAP of classification results using rank minimization for light transport change recovery is still satisfactory.

Apart from the mAP evaluation using one-versus-all classification, we also show the confusion matrix of a 15-way SVM on the testing data in Figure 7. In the confusion matrix, we can see that most cases are correctly classified. In a number of cases, a small group of people is misclassified as a large group of people gathering. This is partially because of the unbalanced training data, and partially because of the difficulty for the light transport model to distinguish the size of a group.

As a comparison, we also investigate whether it is a good practice to normalize features and use RBF kernel SVM for this problem. We repeat the experiments using unnormalized features and linear SVM as classifiers, and report the mAP in Table I. From the table we can see both feature normalization and using RBF kernel SVM improve the classification

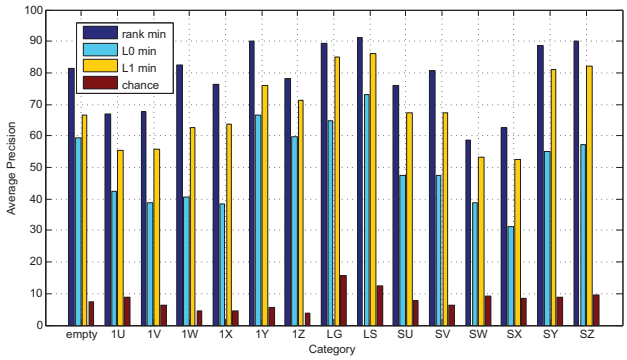

Fig. 6: The average precisions (AP) for the fifteen-category classification problem.

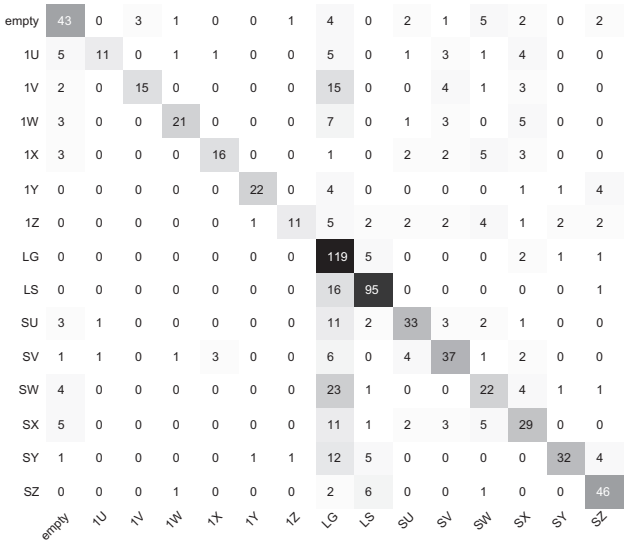

Fig. 7: The confusion matrix of the fifteen-category classification results using rank minimization recovery for light transport changes. Rows are ground truth and columns are predictions.

performance significantly. Feature normalization is important because different sensors, or even different channels of the same sensor may have different sensitivity. Using kernel SVM is important because the classification boudaries in the feature space are nonlinear.

VII. Discussions: Limitations and Future Improvements

During the pre-processing of the data, for each individual or group of subjects, multiple measurements are made, and we partition these measurements to training set and testing set with ratio $6:4$. All the experimental results reported in Section VI use such a data partition protocol. This is good enough for experiments. However, to deploy a practical system, we want our algorithm to be able to generalize to new subjects. In such a problem, the subjects in the testing

TABLE I: The mean average precisions (mAP) of the fifteen-category classification problem using different features, different classifiers and different light transport change recovery methods.

mAP (%)	normalized features, RBF kernel SVM	normalized features, linear SVM	unnormalized features, RBF kernel SVM	unnormalized features, linear SVM
rank minimization	78.69	66.07	67.96	61.27
ℓ_0 minimization	50.76	40.63	34.20	29.34
ℓ_1 minimization	68.37	53.44	55.27	47.52

set should never appear in the training set. We also carried out this experiment for fifteen-category classification, and find out that the mAP performance drops dramatically to 22.09%, 21.73% and 16.33% for rank minimization, ℓ_1 minimization, and ℓ_0 minimization, respectively. This is because our dataset is relatively small, obtained with only nine subjects. The light transport model describes the colors in the space. Since different subjects wear clothes of different colors, a subject generalization task with such a small number of subjects should expect failure. Thus, to implement a practical system in the future, we need to perform the training using a very large dataset, covering all possible clothes colors.

We are using Seachanger Colorbug sensors in our experiments. We point out that these are just the sensors we use for current experiments. These sensors are expensive, have a long response delay, need to be charged every six hours, and receive light from all directions, thus are not suitable for practical systems (some functionalities also overfit our problem). However, it is not difficult to build lightweight color sensors with low cost, quick response and robust performance (*e.g.* using photodiodes). Once we can directly integrate such sensors into the lighting system, it is possible to achieve real time occupancy determination and intelligent control of the room.

VIII. Conclusions

In this paper, we presented a room occupancy determination system using color controllable LED fixtures and color sensors. Based on the inputs to the LEDs and the measurements from the sensors, we recovered the light transport model, and used the model parameters to determine the occupancy in this smart room. The challenge of this problem is that we can only take a limited number of measurements within a short time interval, thus we added sparsity constraints to the recovery problem to get the unique solution. Using features extracted from the recovered light transport matrices, we set up two different classification problems, and achieved high performance. One of our key observations is that the light transport matrix recovered using rank minimization performs significantly better in the classification problem than those recovered by ℓ_0 or ℓ_1 minimization.

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 authors would like to thank Prof. Robert Karlicek for the constructive suggestions and criticism. The authors would also like to thank Dr. Zhenhua Huang, Mr. Sina Afshari, Dr. Li Jia, Mr. Cyril Acholo, Mr. Anqing Liu, Prof. Richard J. Radke and Prof. Sandipan Mishra for the helpful discussions.

Besides, the authors would like to thank Miss Xianqing Zou, Miss Zhen Li, Mr. Rui Ma, Mr. Qingxuan Kong, Mr. Yu Xia, Miss Xia Yang, Miss Kaitai Li and Miss Yiheng Lou for helping with the data collection.

References

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

[2] S. Afshari, S. Mishra, A. Julius, F. Lizarralde, J. D. Wason, and J. T. Wen, "Modeling and control of color tunable lighting systems," *Energy and Buildings*, vol. 68, Part A, no. 0, pp. 242–253, 2014.

[3] L. Jia, S. Afshari, S. Mishra, and R. J. Radke, "Simulation for pre-visualizing and tuning lighting controller behavior," *Energy and Buildings*, vol. 70, no. 0, pp. 287–302, 2014.

[4] L. Jia and R. Radke, "Using time-of-flight measurements for privacy-preserving tracking in a smart room," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 689–696, Feb 2014.

[5] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. Lensch, "Dual photography," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 745–755.

[6] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[7] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[8] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[9] D. L. Donoho, V. C. Stodden, and Y. Tsaig, "About sparselab," 2007. [Online]. Available: http://sparselab.stanford.edu

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[11] E. Candes and J. Romberg, "ℓ_1-magic: Recovery of sparse signals via convex programming," vol. 4, 2005. [Online]. Available: http://users.ece.gatech.edu/~justin/l1magic/

[12] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[13] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 494–504, 2012.

[14] S. Jokar and V. Mehrmann, "Sparse solutions to underdetermined kronecker product systems," *Linear Algebra and its Applications*, vol. 431, no. 12, pp. 2437–2447, 2009.