3D Scene Estimation with Perturbation-Modulated Light and Distributed Sensors

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

***Abstract*—In this paper, we present a framework to roughly reconstruct the 3D occupancy scenario of an indoor space using color-controllable light and distributed color sensors. By applying randomly generated perturbation patterns onto the input of the LED fixtures, and measuring the changes of the sensor readings, we are able to recover the light transport model (LTM) of the room. Then a variant of the inverse Radon transform is applied on the LTM to reconstruct the 3D scene. The reconstructed scene by our algorithm can faithfully reveal the occupancy scenario of the indoor space, while preserving the privacy of human subjects. An occupancy-sensitive lighting system can be designed based on this technique.**

***Keywords*-occupancy scenario; controllable light; light transport model; inverse Radon transform; 3D reconstruction**

I. Introduction

In traditional vision, cameras and depth sensors are widely used to capture images, videos, and depth maps of a scene. An image, whether gray-level, RGB, or depth, has a 2D structure, which describes the spatial distribution of the scene. A lot of high-level information can be inferred from such data, which enables numerous applications such as object tracking, event detection, and traffic surveillance. However, in some applications, human-readable high-resolution images are not only unnecessary, but also an information security concern. For example, if we want to monitor the occupancy of a room for the task of intelligent lighting control, we only need a very rough estimation about which part of the room is occupied. Using cameras will raise the concern of privacy — people just feel uncomfortable being monitored by a camera, not to mention that the security of camera networks can be compromised.

In such cases, low-level sensors could be good alternatives. Also, these sensors are generally much cheaper than cameras. However, the output of a low-level sensor is usually only a few numeric values, rather than a 2D-structured image. This makes it very difficult to infer high-level information. Reconstructing a 3D scene from these 1D signals is an extremely ill-posed and challenging problem.

However, if the light can be controlled, and color sensors are employed, we are able to measure the sensor outputs under different lighting conditions. With repeated measurements, we are able to construct a model for the spatial transport of the light. Such a model captures rich information of the 3D space, and can be used to roughly estimate the 3D scene. Once we are able to reconstruct the 3D scene, we can use this information to monitor and control the environment. We are particularly interested in the intelligent control of energy-saving and occupancy-sensitive light delivery.

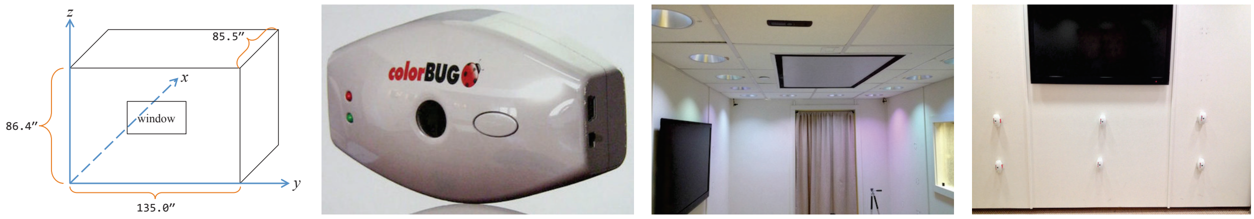
II. Testbed Setup

A. The Smart Room

To validate this idea, we established a smart lighting room as our testbed. This room has one window and two doors, and is 85.5 inches wide, 135.0 inches long, and 86.4 inches high (Figure 1a). This testbed is equipped with twelve color controllable LED light fixtures mounted in the ceiling (Figure 1c). For each fixture, we can specify 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 1b) on the walls of this room (Figure 1d), six on each side. 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 the color sensors with Wi-Fi, and sends input signals to the fixtures via Bluetooth. This same smart room has been used for a number of other investigations, including lighting control algorithms [2], [3], [4] and visual tracking systems [5].

B. The Occupancy-Sensitive Lighting System

The final goal of our system is to achieve occupancy-sensitive smart lighting. In other words, when the occupancy of the room changes, the system should produce the lighting condition that best suits this occupancy scenario to maximize comfort and minimize energy consumption. Here by "occupancy" we mean the number and spatial distribution of people in the room. For this purpose, there should be a control strategy module and a scene estimation module, and they work with two alternating stages: the *sensing stage* and the *adjustment stage* (Figure 3). In the sensing stage, the scene estimation module collects the sensor readings to estimate the scene; in the adjustment stage, the control strategy module decides what lighting condition should be produced based on the reconstructed scene. How to design the control strategies is beyond the scope of this paper. Here we focus on the scene estimation module.



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

Figure 1: The testbed setup. (a) The coordinate system of the room. (b) The Seachanger wireless Colorbug sensor. (c) Twelve color controllable LED fixtures illuminate the room from the ceiling. (d) Color sensors are placed on the walls.

III. Light Transport Model

Since the current configuration of our testbed has twelve LED fixtures of 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)* introduced in [6]. Here we call x_0 the *base light*, which is determined by the control module.

IV. Perturbation-Modulated Measurement

A. Solve for A

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. In other words, we perturb the input to 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 $Y = AX$. If we make many measurements to ensure $n > m_1$, then this overdetermined linear system can be solved by the pseudo-inverse:

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

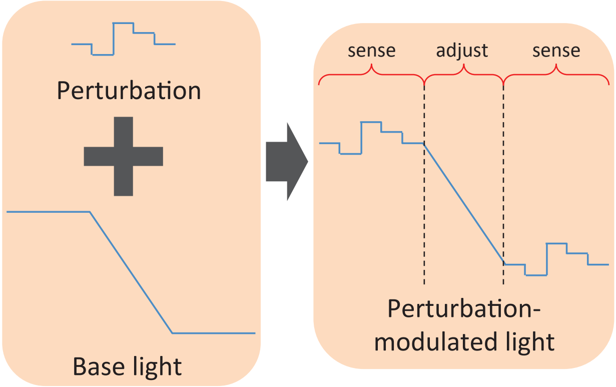


Figure 2: The concept of perturbation-modulated measurement.

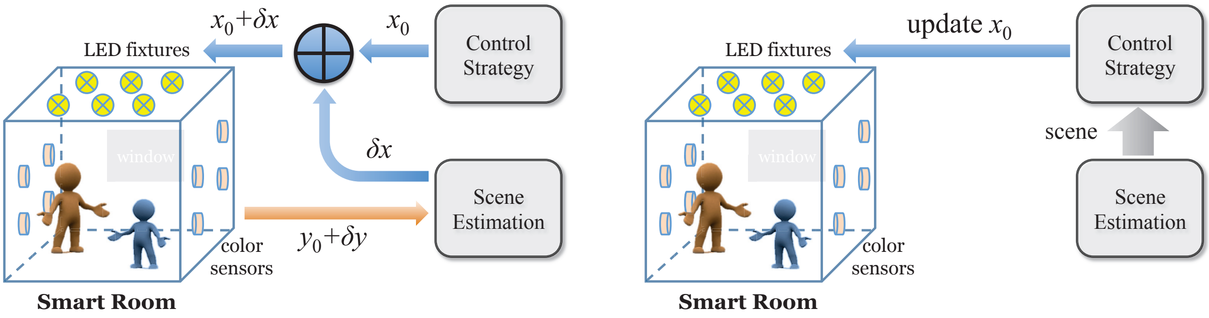
B. Perturbation Modulation

As introduced in Section II-B, the smart lighting system works with two alternating stages: sensing and adjustment. During the sensing stage, perturbations δx are added to the base light x_0, and δy is measured. Then in the adjustment stage, matrix A is computed, the scene is reconstructed, and the control module gradually changes the base light to a new one, which is determined according to the reconstructed scene. In such a system, the base light changes slowly within a large range, while the perturbation changes quickly, and ideally imperceptibly, within a small range (Figure 2). This is analogous to amplitude modulation (AM) in electronic communications, where low-frequency information rides on a high-frequency carrier. The difference is that, in our system, the intensity of the carrier, or the base light, is the low-frequency component.

C. Requirements for Perturbation Patterns

For accurate recovery of the light transport matrix and the comfort of human subjects, we identify three requirements on the perturbation patterns:

1) The perturbation patterns must be rich in variation to capture sufficient information from the scene.



(a) sensing

(b) adjustment

Figure 3: Two stages of the lighting system. (a) In the sensing stage, the scene estimation module measures sensor outputs under different lighting conditions. (b) In the adjustment stage, the control module uses the reconstructed scene to determine what base light should be produced.

2) The magnitude of the perturbation must be small enough not to bother humans in the room.
3) The magnitude of the perturbation must be large enough to be accurately measured by the color sensors.

To meet the first requirement, randomly generated patterns are usually good enough. If we define the magnitude of the perturbations as $\rho = \max_i ||\delta x_i||_\infty$, then the choice of ρ is a trade-off. In our work, we set $\rho = 0.025$ (based on a range of $[0, 1]$) such that perturbations are not easily noticed by human subjects, but are easily sensed by our current color sensors.

V. Reconstruction Algorithm

Let the light transport matrix of an empty room be A_0. At run time, the light transport matrix is A, and we call $E = A_0 - A$ the *difference matrix*. Matrix E is also $m_2 \times m_1$, and each entry of E corresponds to one fixture channel and one sensor channel. If one entry of matrix E has a large positive value, it means that the light paths from the corresponding fixture to the corresponding sensor are very likely blocked. From any given fixture to any given sensor, there are numerous diffuse reflection paths and one direct path, which is the line segment connecting the fixture and the sensor (Figure 4a). Obviously, the direct path is the dominating path, if one exists. Thus a large entry of E may most likely imply the corresponding direct path has been blocked by the change of occupancy.

A. Aggregation of E

Though each entry of E corresponds to one direct path, the opposite is not true, since each LED fixture or sensor has multiple channels. Assume the number of LED fixtures is N_L, and the number of sensors is N_S. We aggregate the $m_1 \times m_2$ matrix E to an $N_S \times N_L$ matrix $\hat{E}$, such that the mapping from the entries of $\hat{E}$ to all direct paths is a bijection. In our experiments, $m_1 = 3N_L = 36$ and $m_2 = 4N_S = 48$. The aggregation is performed on each fixture-sensor pair as a summation over three color channels: red-to-red, green-to-green, and blue-to-blue (Figure 5), or simply:

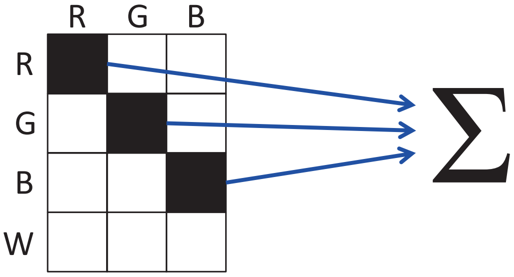


Figure 5: Aggregating the difference matrix E to $\hat{E}$ on same channels. This diagram shows the aggregation on a 4×3 submatrix of E.

$$\hat{E}_{i,j} = E_{4i-3,3j-2} + E_{4i-2,3j-1} + E_{4i-1,3j}. \tag{4}$$

B. Volume Rendering

After aggregation, now if $\hat{E}$ has a large entry at (i, j), then we believe the direct path from fixture j to sensor i is very likely blocked, though we are still not sure where the blockage happens on this path. By assuming the occupancy is continuous and smooth, we also believe that any position that is close to this direct path is also likely being occupied. If two or more such direct paths intersect or approximately intersect in the 3D space, then it is most likely that the blockage happens at their intersection, as shown in Figure 4b.

Based on this philosophy, we now describe our reconstruction algorithm. Let P be an arbitrary point in the 3D space, and $d_{i,j}(P)$ be the point-to-line distance from point P to the direct path from fixture j to sensor i. The confidence

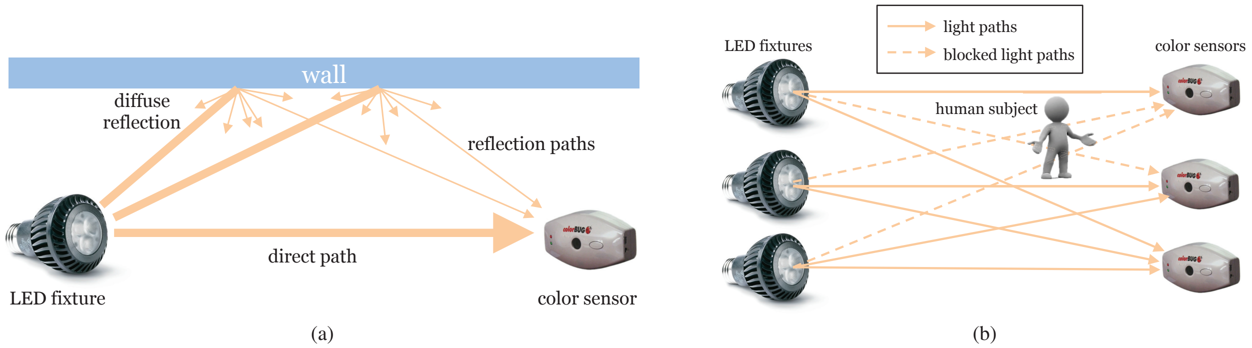


Figure 4: (a) Light paths from one fixture to one sensor. (b) Intersecting blocked light paths imply blockage at the intersection.

of point P being occupied is $C(P)$, which is computed by:

$$C(P)=\frac{\sum_{i=1}^{N_S}\sum_{j=1}^{N_L}\hat{E}_{i,j}G\left(d_{i,j}(P),\sigma\right)}{\sum_{i=1}^{N_S}\sum_{j=1}^{N_L}G\left(d_{i,j}(P),\sigma\right)}, \tag{5}$$

where $G(\cdot,\cdot)$ is the Gaussian kernel:

$$G(a,\sigma)=\exp\left(-\frac{a^2}{2\sigma^2}\right). \tag{6}$$

The denominator in Eq. (5) is a normalization term for the non-uniform spatial distribution of the LED fixtures and the sensors. The σ is a measure of the continuity and smoothness of the occupancy. If we discretize the 3D space and evaluate Eq. (5) at every position $P(x,y,z)$, we are able to render a 3D volume $V(x,y,z)=C(P(x,y,z))$ of the scene, which is also called the 3D *confidence map*.

C. Connection with Radon Transform

Our reconstruction method is partially inspired by the well-known Radon transform, or more precisely, the inverse Radon transform, which has been successfully applied to the reconstruction of computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), single photon emission computer tomography (SPECT), and even radar astronomy [7], [8]. Given a continuous function $f(x,y)$ on $\mathbb{R}^2$, its Radon transform $\mathscr{R}f$ is a function defined on each straight line $L=\{(x(t),y(t))\}$ in $\mathbb{R}^2$:

$$\mathscr{R}f(L)=\int_L f(x(t),y(t))dt. \tag{7}$$

Since a straight line can be uniquely defined by two parameters, $\mathscr{R}f$ is also a function on $\mathbb{R}^2$. The original function f can be reconstructed by the inverse Radon transform, which comprises a ramp filter and a back-projection. An example is shown in Figure 6. In our reconstruction algorithm Eq. (5), the denominator corresponds to the ramp filter, and the summation over all direct light paths corresponds to the back-projection.

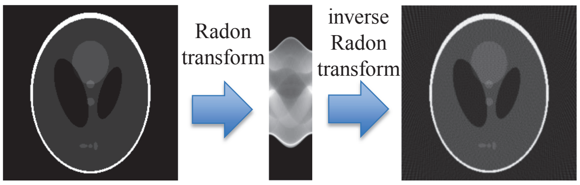


Figure 6: Radon transform on the Shepp-Logan phantom [9].

In computed tomography, multiple X-ray sources and sensors can be rotated around the object to create numerous lines, and 3D images can be acquired slice by slice. However, in our problem, with only twelve LED fixtures and twelve sensors, the light paths are very sparse (Figure 8), which makes reconstruction much more challenging than other problems that could be solved by a standard Radon transform.

VI. Experimental Results

To validate our method, we divide the smart room into six regions, and create different occupancy scenarios by occupying one or two regions with human subjects and furniture. We discretize the 3D space to voxels of size $1\times1\times1$ inch^3, and render 3D volumes of size $87\times136\times88$. For the Gaussian kernel we set $\sigma=20.0$ inches. In the sensing stage, $n=40$ perturbation patterns are used.

A. Reconstruction Results

In Figure 7 we show the reconstruction results for scenarios where one or two regions are occupied. It is interesting to see that although the precision of the reconstructed volume is very low, the reconstruction quality is good enough for the control module to determine which part of the room is occupied, and what kind of light should be delivered. If better reconstruction quality is required, one simple solution is to increase the number of color sensors.

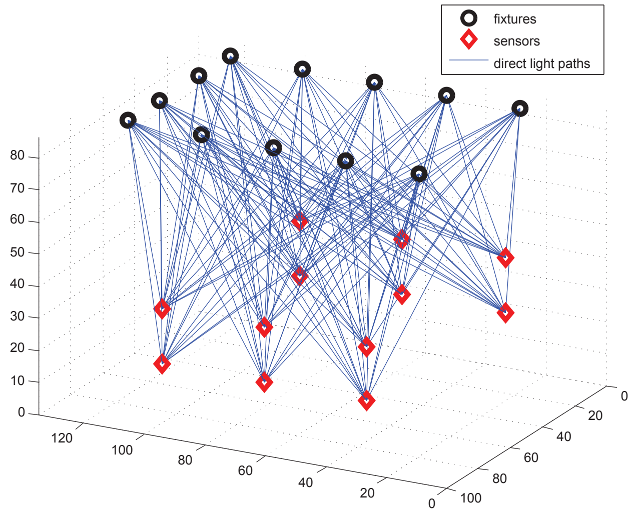


Figure 8: 144 direct light paths in the smart room.

B. Complexity Analysis

Assume the number of voxels in one volume is N_P. The number of direct light paths is $N_L \cdot N_S$. To render one volume, we have to evaluate Eq. (5) for N_P voxels, and the number of operations is $N_P \cdot N_L \cdot N_S$ in total. In one operation, we need to compute the point-to-line distance and the Gaussian kernel. In our experiments, $N_P = 87 \times 136 \times 88$, $N_S = 12$, and $N_L = 12$. Thus the number of operations is about 150 million. Our rendering algorithm is implemented in C++. On a Macintosh with 2.5 GHz Intel Core i5 CPU and 8 GB memory, the direct algorithm takes about 18 seconds to render one volume.

C. Accelerations

One way to accelerate the rendering is to pre-compute the point-to-line distances and the Gaussian kernels, and keep them in memory. When rendering a new volume, we still need to perform $N_P \cdot N_L \cdot N_S$ operations, but each operation is simply one multiplication and one addition. In this way, on the same machine, pre-computation takes about 18 seconds, but rendering one volume takes only 2 seconds. One trade-off is that such a hashing-based optimization uses much more memory. If each Gaussian kernel is stored as a 64-bit double-precision floating point number, then it requires about 1 GB memory to keep 150 million Gaussian kernels. To further accelerate the rendering to achieve real-time performance, parallel computing on a GPU could be used.

VII. Conclusions

We have presented a 3D scene estimation framework based on perturbation-modulated light and distributed color sensors. This 1D signal to 3D volume problem is extremely ill-posed and very challenging. By recovering the light transport model from the sensor output changes under randomly perturbed light, we successfully applied an inverse Radon transform alike algorithm to roughly reconstruct the 3D scene. The reconstructed occupancy maps suffice for intelligent control of light delivery, but are coarse enough to protect the privacy of human subjects. Though the experiments are carried out with visible light and color sensors, it is very promising to generalize this framework to other types of light and sensors beyond the visible spectrum.

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.

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, and J. T. Wen, "Modeling and feedback control of color-tunable led lighting systems," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 3663–3668.

[3] 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.

[4] 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.

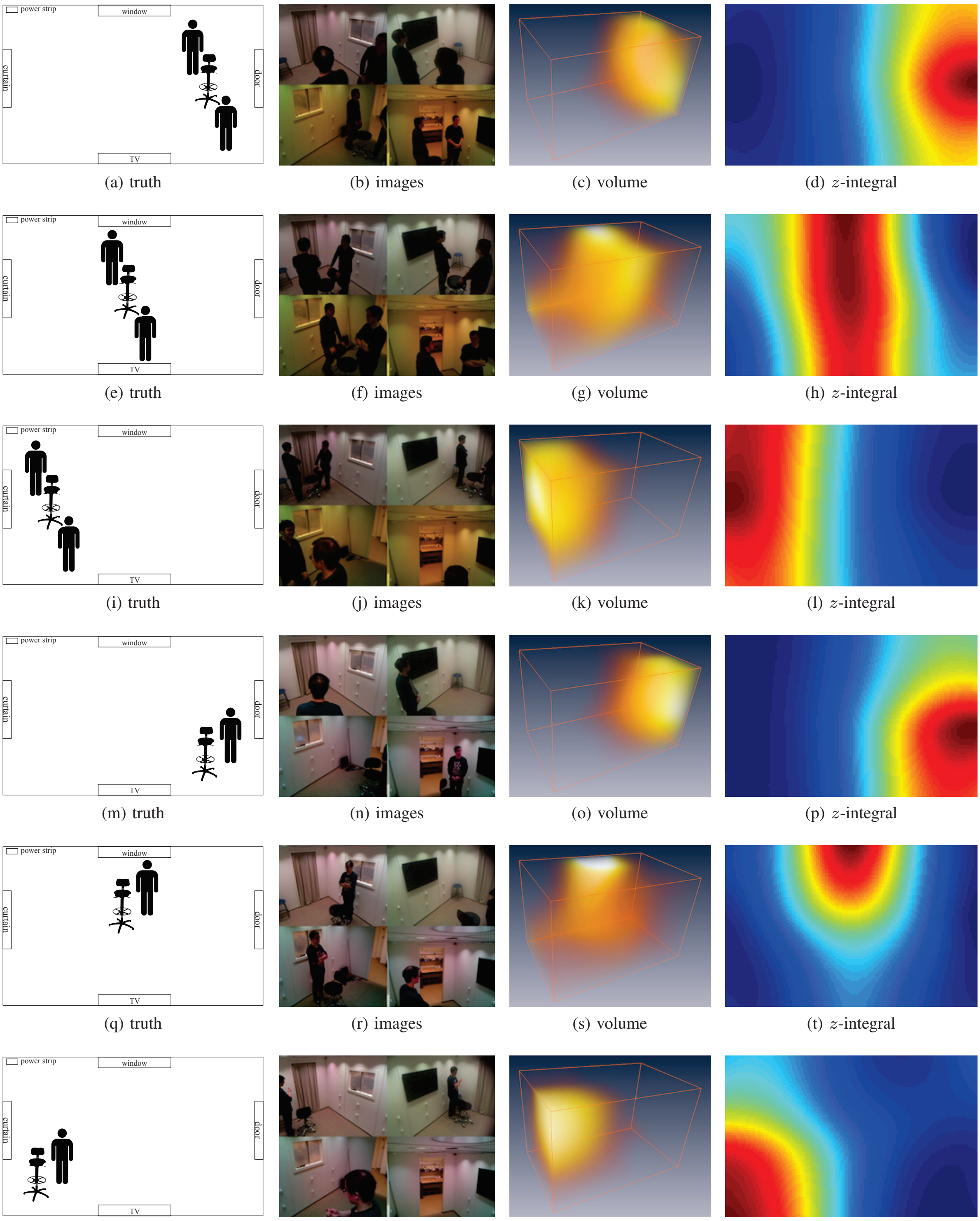
[5] L. Jia and R. Radke, "Using time-of-flight measurements for privacy-preserving tracking in a smart room," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 689–696, Feb 2014.

[6] 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.

[7] J. Radon, "On determination of functions by their integral values along certain multiplicities," *Ber. der Sachische Akademie der Wissenschaften Leipzig,(Germany)*, vol. 69, pp. 262–277, 1917.

[8] S. R. Deans, *The Radon transform and some of its applications*. Courier Dover Publications, 2007.

[9] L. A. Shepp and B. F. Logan, "The fourier reconstruction of a head section," *Nuclear Science, IEEE Transactions on*, vol. 21, no. 3, pp. 21–43, 1974.



(a) truth (b) images (c) volume (d) z-integral

(e) truth (f) images (g) volume (h) z-integral

(i) truth (j) images (k) volume (l) z-integral

(m) truth (n) images (o) volume (p) z-integral

(q) truth (r) images (s) volume (t) z-integral

(u) truth (v) images (w) volume (x) z-integral

Figure 7: Reconstruction results for different occupancy scenarios. Each row is one scenario. Column 1: a diagram of the ground truth scenario; Column 2: images captured by four cameras in the room during measurement; Column 3: the reconstructed 3D volume; Column 4: the integral of the reconstructed volume on z-axis, to be compared with the ground truth.